Bangladesh University of Professionals

Faculty of Science and Technology
Department of Information & Communication Technology (ICT)

Course Title: Database Management System Laboratory
Course Code: ICE-2204

Project Report

Project Name: "EduSync: Streamlined Student Administration Platform"

Submitted to

Sharmeen Jahan Seema	Afrina Khatun
Lecturer, Dept. of ICT Faculty of Science & Technology Bangladesh University of Professionals	Assistant Professor, Dept. of ICT Faculty of Science & Technology Bangladesh University of Professionals

Submitted by

Section – A, Batch: BICE – 2022

Gazi Mohammad Abrar Zawad	2254901049
Jablay Noor Rahman	2254901093
Prithviraj Chowdhury	2254901101

Submission Date: 02 – 06 – 2024

Table of Contents

Phase-01

Introduction 3

Summary 3

Functions & Operations 4

1.Index/Home Page 4

2. User:Login Page 4

3.Registration/Signup Page 5

4.Student Registration 5

5.Faculty Registration 6

6.Parent Registration 6

7.Student Dashboard 7

8.Student: View Course 7

9.Student: View Attendance 8

10.Student: View Marks 8

11.Student: View Notices 9

12.Faculty Dashboard 9

13.Faculty: Give Attendance 10

14.Faculty: Update Attendance 10

15.Faculty: Provide Marks 11

16.Faculty: Add New Courses 11

17.Faculty: View Notices 12

18.Parent: Dashboard 12

19.Parent: View Attendance 13

20.Parent: View Marks 13

21.Parent: View Notices 13

22.Parent: See Statistics 14

23.Parent: Download Report PDF 14

24.Admin: Dashboard 15

25.Admin: Add Courses 16

26.Admin: Delete Courses 16

27.Admin: Update Courses 17

28.Admin: Add Department 17

29.Admin:Delete Department 18

30. Admin: Add Notices 18
31.Admin: Delete Notices 18
32.Admin: View All Notices 19
33.Admin: Attendance Analysis 19
34.Admin: View Course Performance 19
35.Admin: User Statistics 20
36.Admin: See Patterns/Trends: 21
37.Admin: Generate Reports 23
Entity Relationship Diagram (Updated) 24
Narrative Description of ER-Diagram 25
Progress Report: 27
Summary 27
Group Progress Report 28
Individual Report 29
Phase-02 .....
1.Schema Refinement 32
Integrity Constraints 33
2.Normalization 34
3.SQL Statements 43
4.Populating the Database 49
5.Pattern Identification Queries 54
6.Progress Reports 56
Learning Outcomes of the Project 62
Practical Applications of the Project 63
Limitations of the Project 64
Conclusion 65

Phase-01: Introduction

EduSync: Streamlined Student Administration Platform is designed to streamline the administrative tasks and provide a unified interface for students, faculty, parents, and administrators within an educational institution. The purpose of this project is to simplify the process of managing academic and administrative tasks, enhancing the efficiency and effectiveness of the institution's operations. EduSync aims to improve the current system, making it more accessible and efficient for all users. This platform will enable better communication, data management, and reporting capabilities.

Summary

The EduSync project aims to develop an integrated educational management system designed to enhance the efficiency and effectiveness of academic and administrative tasks within an educational institution. The system provides distinct dashboards and functionalities tailored for students, faculty, parents, and administrators, each addressing their specific needs and roles.

Students: Upon logging in, students are directed to their personalized dashboard, where they can access detailed information about their courses, view their marks for various exams and assignments, check their attendance records, and stay updated with the latest institutional notices. Additionally, students can view attendance records for all courses in their department and check attendance for any course on a particular date.

Faculty: Faculty members have a dedicated dashboard that allows them to manage their teaching responsibilities efficiently. They can take attendance for their courses, update pre-existing attendance records, and input marks for student exams and assignments. Faculty members can also add new courses with detailed descriptions and schedules according to their preferences. Furthermore, they can view important notices announced by the administrator.

Parents: Parents are provided with a dashboard where they can monitor their children’s academic and attendance performance. They can view detailed attendance records, including the total number of classes for a course and the attendance percentage. Parents can also see their children's marks and the highest marks for course exams, stay informed with the latest announcements, and visualize and download statistical data of their child’s academic and attendance performance.

Admins: Administrators have the most comprehensive set of functionalities to manage the institution's operations. From their dashboard, admins can add, update, and delete courses, create and remove departments, and manage notices by adding, deleting, and viewing all notices. They can generate detailed reports and statistical analyses based on the available data, providing valuable insights into the institution's performance. Admins also have

access to detailed performance/pattern metrics for individual students and faculty members, enabling them to make informed decisions.

Functions & Operations

Below are the primary functions along with their descriptions that will be included in the project.

1.Index/Home Page:

The index or home page of EduSync Platform that will contain the logo in the middle where the existing users can login and new users can signup for this platform.

Fig1: Index/Home Page

2. User: Login Page

Description: The login page for users will require all users to fill in the username and password fields. If the username or password is incorrect, the login will be unsuccessful. Once the correct information is entered, the home page for the respective user role will be displayed.

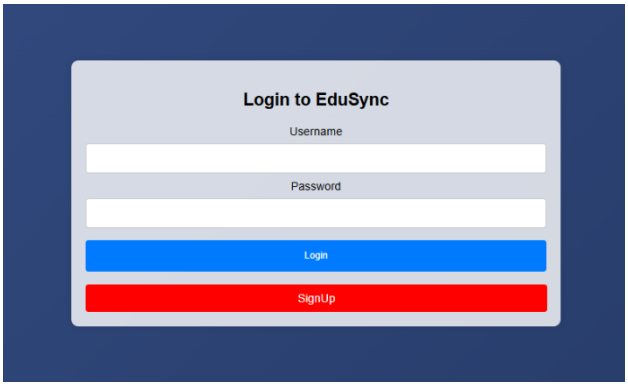


Fig2: Login Page

Registration/Signup

3. Registration/Signup

Description: New users can register on the platform by providing necessary information such as username, password, email, and role. Once registered, users can log in to the platform with their credentials.

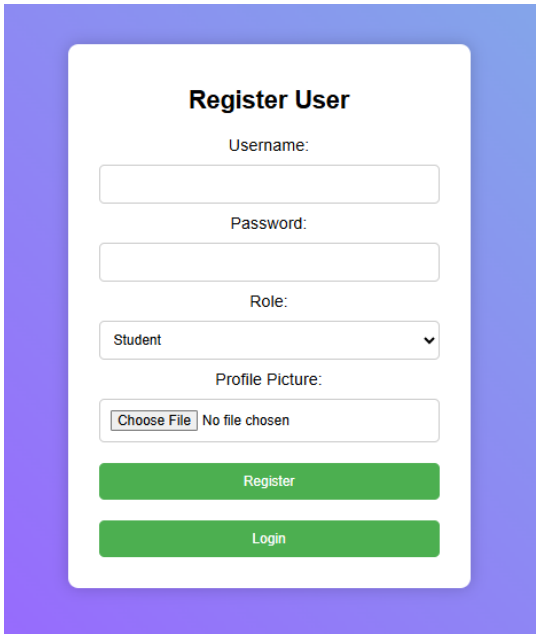


Fig3: Registration Page

4. Student Registration

Description: New students can register on the platform by providing necessary information such as name, enrollment status, department, and contact details. Once registered, students can access their dashboard and other functionalities.

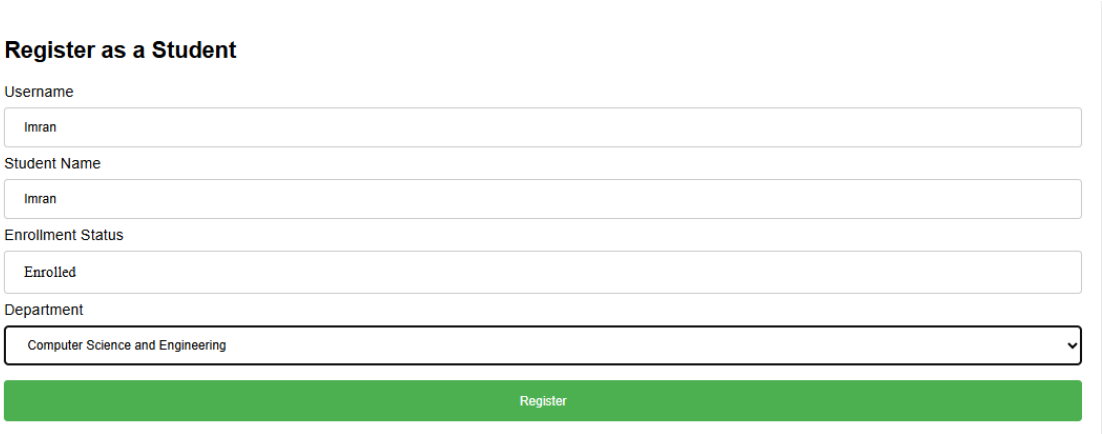


Fig4: Student Registration Page

5. Faculty Registration

Description: New faculty members can register on the platform by providing necessary information such as name, department, and contact details. Once registered, faculty members can access their dashboard and other functionalities.

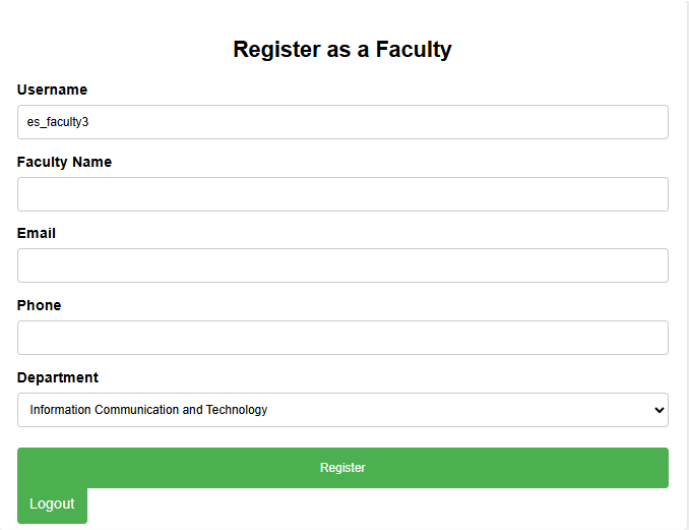


Fig5: Faculty Registration Page

6. Parent Registration

Description: New parents can register on the platform by providing necessary information such as name, their child's student ID, and contact details. Once registered, parents can access their dashboard and other functionalities.

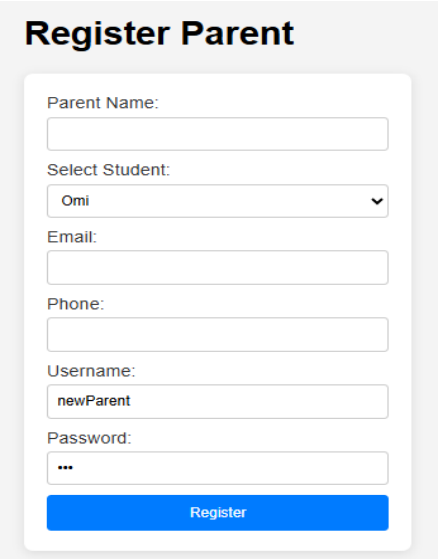


Fig6: Register Parents Page

Students

7. Student: Dashboard

Description: After logging in, students can access their dashboard, where they can view a summary of their academic and attendance information. The dashboard provides a quick overview of their performance and upcoming activities.

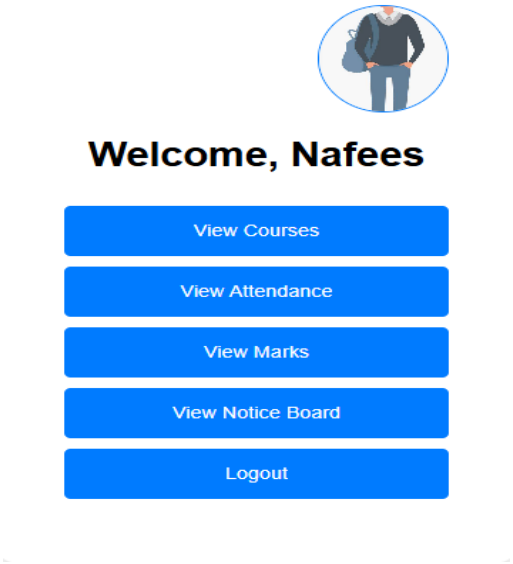


Fig7: Student Dashboard Page

8. Student: View Courses

Description: Students can view the details of the courses they are enrolled in. This includes course names, descriptions, and schedules.

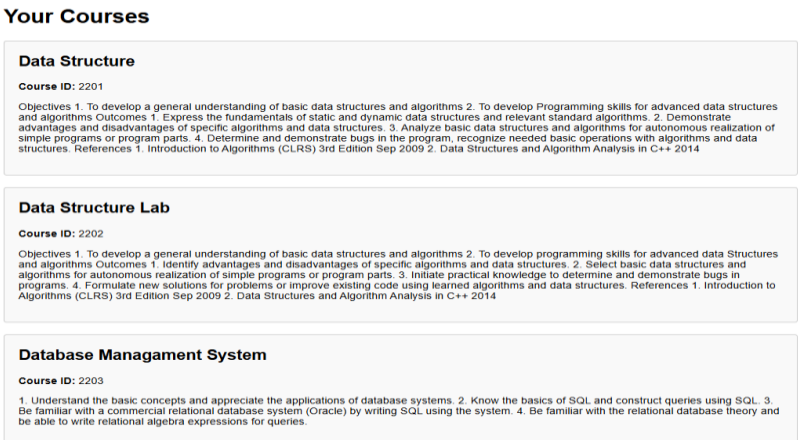


Fig8: Student View Courses Page

9. Student: View Attendance

Description: Students can view their attendance records for each course. This section will show attendance status (Present/Absent) and dates of attendance. Students can also view all attendance records of any course in their department and see attendance for a specific course on a particular date.

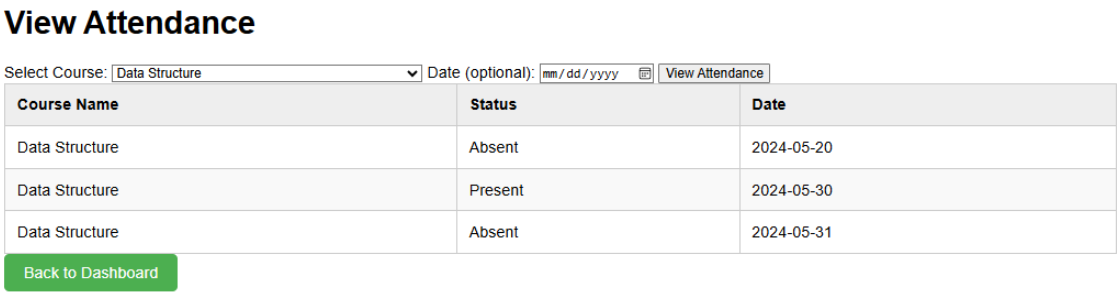


Fig9: Student View Attendance Page

10. Student: View Marks

Description: Students can view their marks for various exams and assignments. This section will display detailed information about their grades, including the exam type, total marks, and obtained marks.

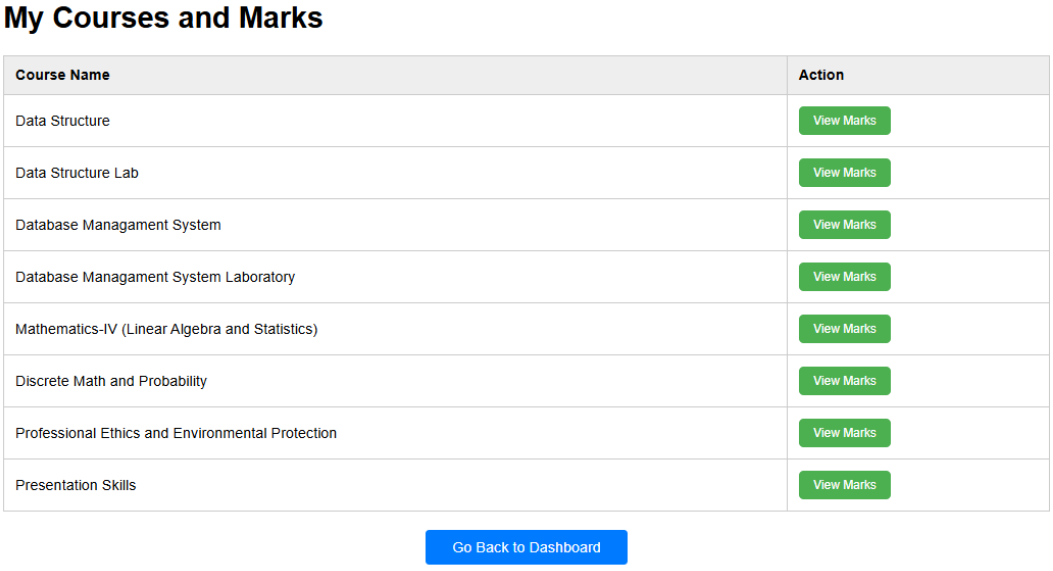


Fig10.1: Student View Marks Page

Fig.10.2: Student View Course Mark Page

11. Student: View Notices

Description: Students can view all notices relevant to them. This section will display important announcements and updates from the institution.

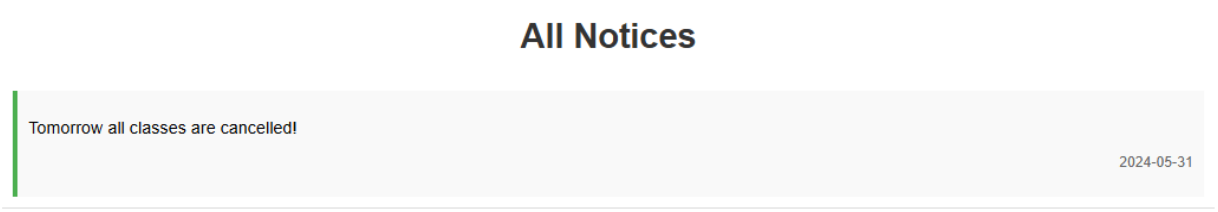


Fig.11: Student View Notices Page

Faculty

12. Faculty: Dashboard

Description: After logging in, faculty members can access their dashboard, where they can view a summary of their courses and administrative tasks. The dashboard provides an overview of their teaching schedule and pending tasks.

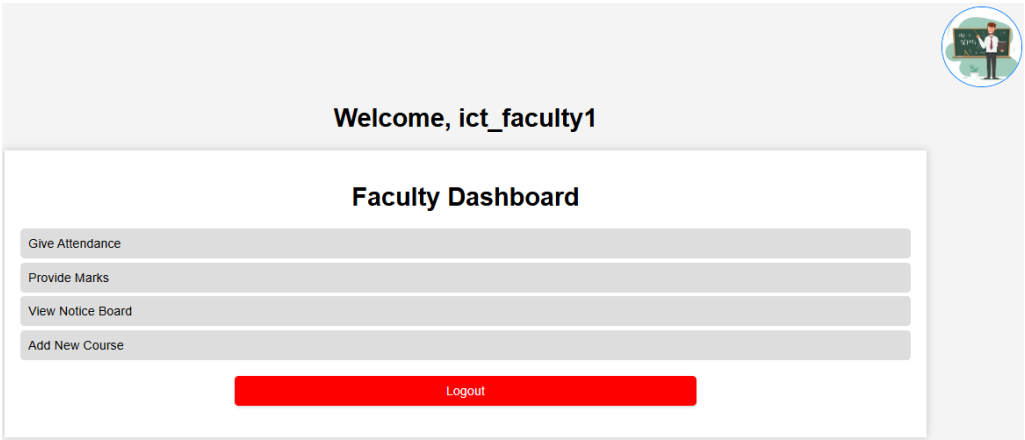


Fig.12: Faculty Dashboard Page

13. Faculty: Give Attendance

Description: Faculty members can record attendance for students in their courses. This function allows them to mark students as present or absent for each class session.

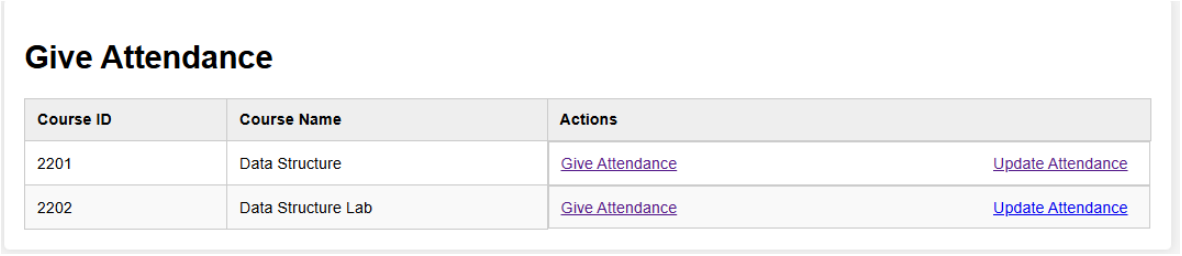


Fig.13.1: Faculty Give Attendance Page

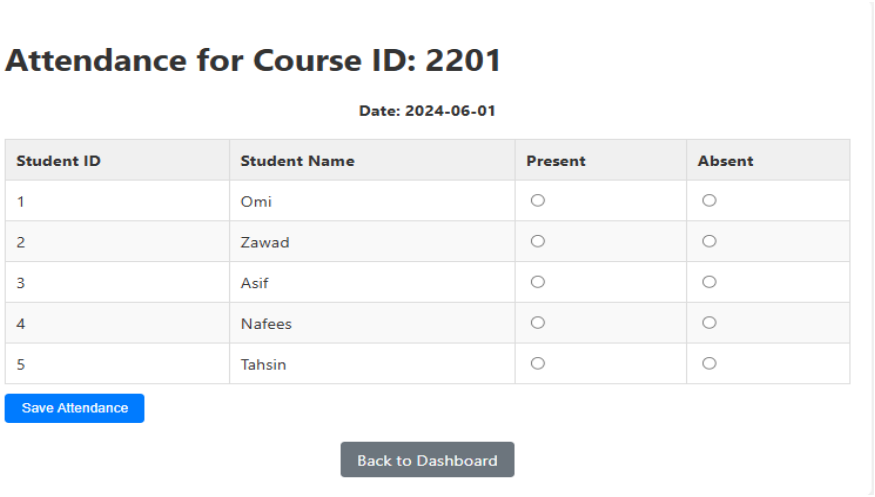


Fig.13.2: Faculty Manage Attendance Page

14. Faculty: Update Attendance

Description: Faculty members can update pre-existing attendance records. This function allows them to correct or modify attendance entries as needed.

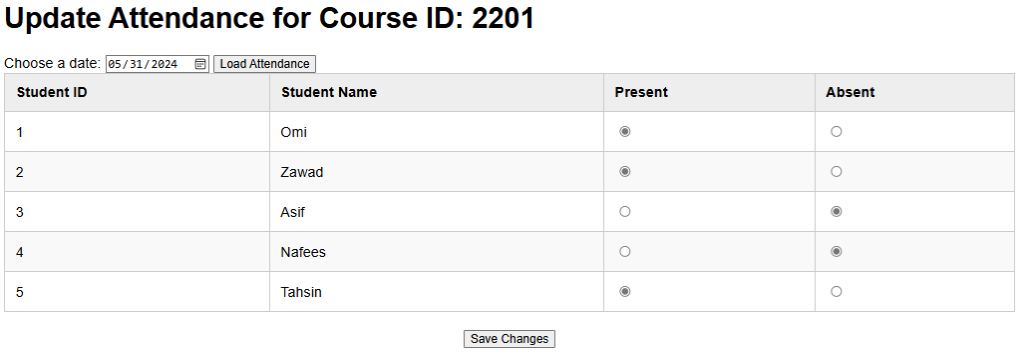


Fig14: Faculty Update Attendance Page

15. Faculty: Provide Marks

Description: Faculty members can input marks for students' exams and assignments. This function allows them to enter grades for each student.

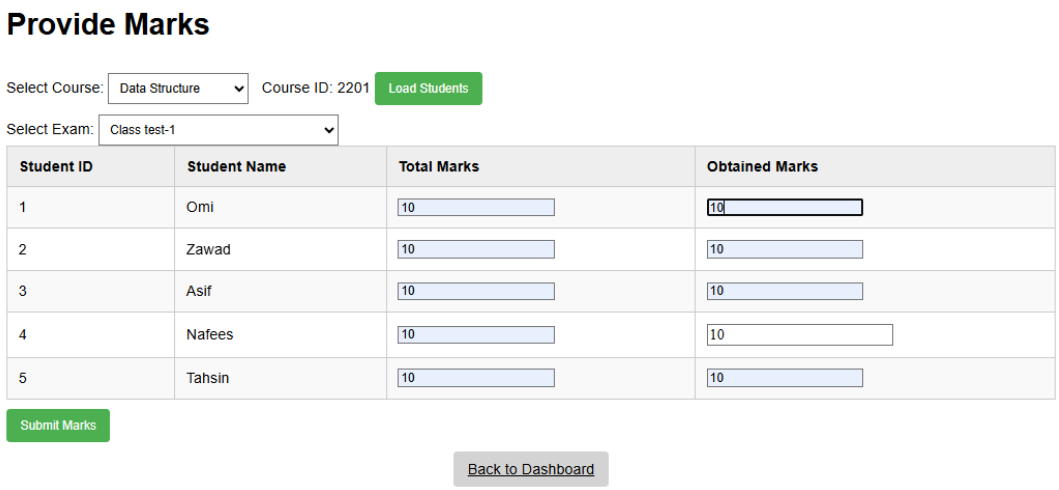


Fig.15: Faculty Provide Marks Page

16. Faculty: Add New Courses

Description: Faculty members can add new courses according to their preferences. This function allows them to create and manage course information, including course names, descriptions, and schedules.

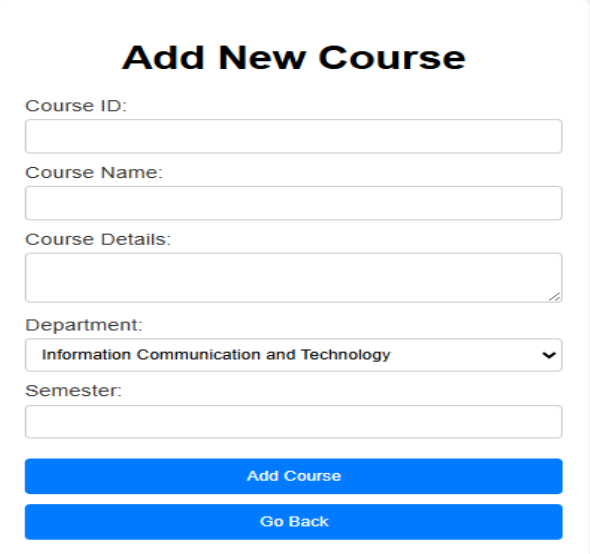


Fig.16: Faculty Add New Courses Page

17. Faculty: View Notices

Description: Faculty members can view all notices announced by the administrator. This section will display important announcements and updates relevant to their teaching responsibilities.

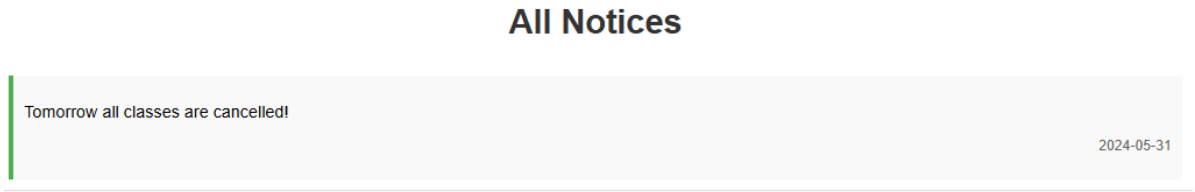


Fig.17: Faculty View Notices Page

Parents

18. Parent: Dashboard

Description: After logging in, parents can access their dashboard, where they can view a summary of their children's academic and attendance information. The dashboard provides a quick overview of their performance and important updates.

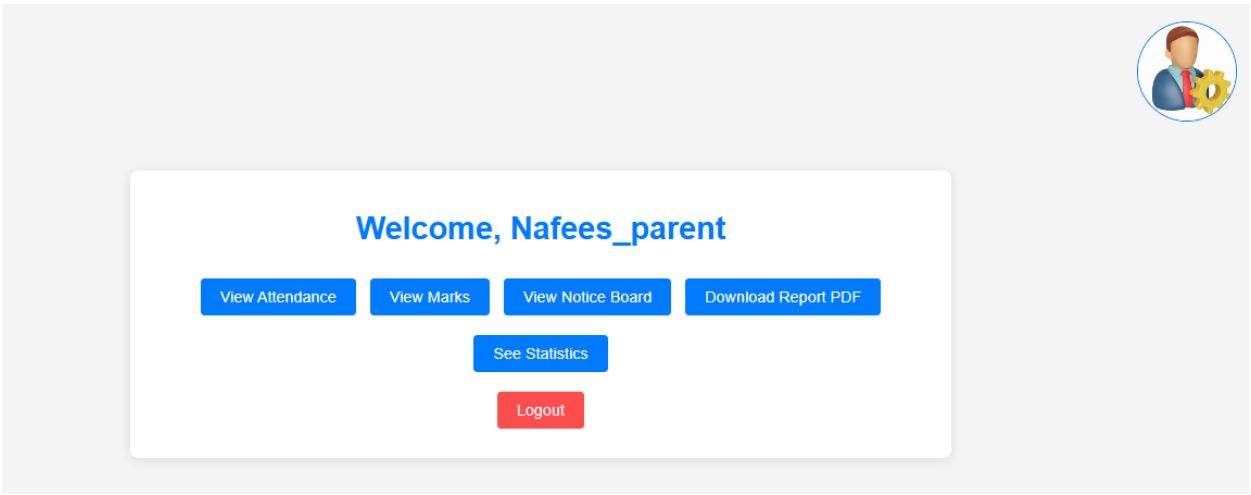
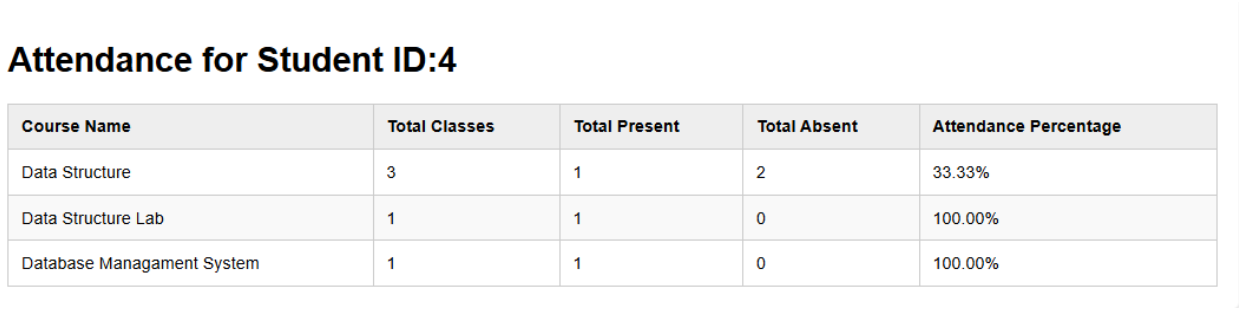


Fig.18: Parents Dashboard Page

19. Parent: View Attendance

Description: Parents can view the attendance records of their children. This section shows the attendance status (Present/Absent) and dates for each class session. Parents can also see the total number of classes for a course and the attendance percentage.



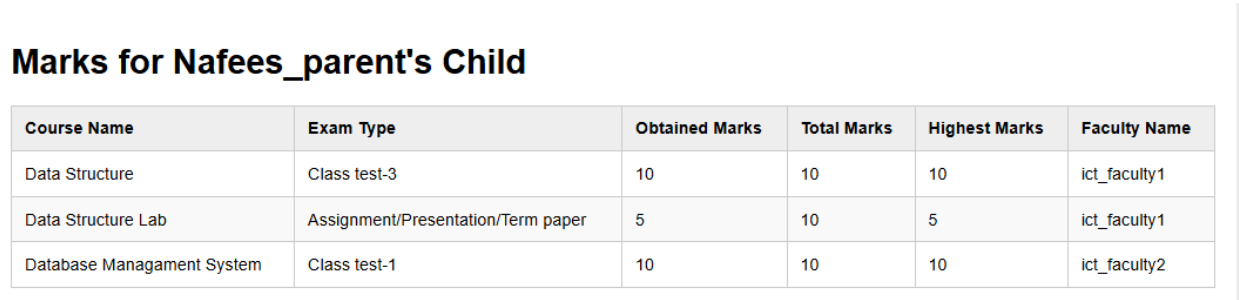
Attendance for Student ID:4

Course Name	Total Classes	Total Present	Total Absent	Attendance Percentage
Data Structure	3	1	2	33.33%
Data Structure Lab	1	1	0	100.00%
Database Managament System	1	1	0	100.00%

Fig.19: Parent View Attendance Page

20. Parent: View Marks

Description: Parents can view the marks of their children. This section displays detailed information about their grades, including the exam type, total marks, and obtained marks. Parents can also see the highest marks of a course exam.



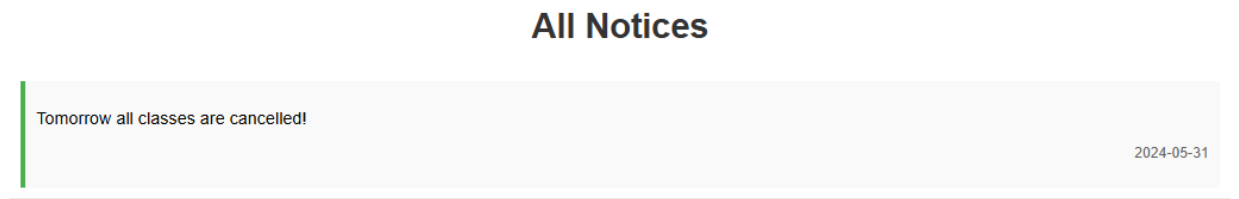
Marks for Nafees_parent's Child

Course Name	Exam Type	Obtained Marks	Total Marks	Highest Marks	Faculty Name
Data Structure	Class test-3	10	10	10	ict_faculty1
Data Structure Lab	Assignment/Presentation/Term paper	5	10	5	ict_faculty1
Database Managament System	Class test-1	10	10	10	ict_faculty2

Fig.20: Parent View Marks Page

21. Parent: View Notices

Description: Parents can view all notices relevant to their children. This section displays important announcements and updates from the institution.



All Notices

Tomorrow all classes are cancelled!

2024-05-31

Fig.21: Parent View Notices Page

22. Parent: See Statistics

Description: Parents can visualize statistics of their child's academic and attendance performance. This function provides graphical representations of their child's performance over time.

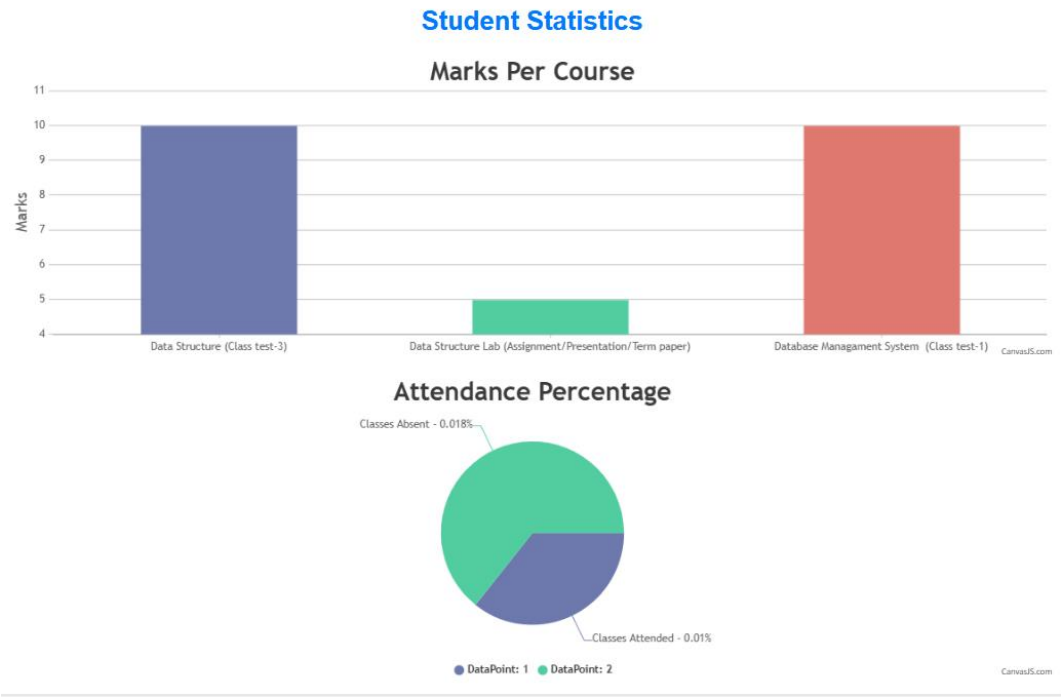


Fig.22: Parent See Statistics Page

23. Parent: Download Report PDF

Description: Parents can download statistics of their child's academic and attendance performance. This function allows them to save the performance data for offline review.

Student Academic and Attendance Report

Parent: Nafees_parent

Student ID: 4

Academic Report

Course	Exam Type	Obtained Marks	Total Marks	Highest Marks
Data Structure	Class test-3	10	10	10
Data Structure Lab	Assignment/Presentation/Term paper	5	10	5
Database Managament System	Class test-1	10	10	10

Attendance Report

Course	Total Classes	Classes Attended	Attendance (%)
Data Structure	3	1	33.33%
Data Structure Lab	1	1	100.00%
Database Managament System	1	1	100.00%

Fig.23: Parent Download Report PDF

Admins

24. Admin: Dashboard

Description: After logging in, admins can access their dashboard, where they can view a summary of the institution's operations, including course management, department management, and administrative tasks. The dashboard provides an overview of pending tasks and important updates.

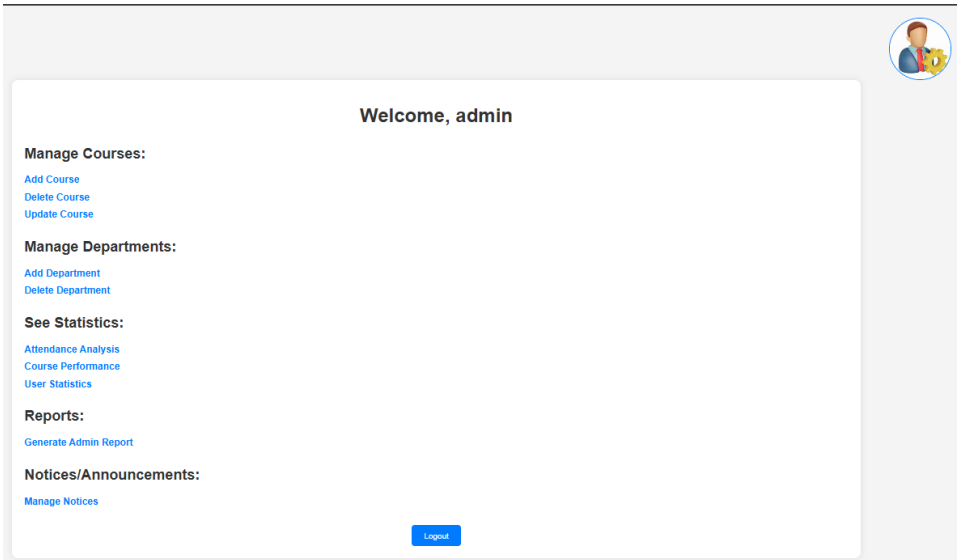


Fig.24: Admin Dashboard Page

25. Admin: Add Courses

Description: Admins can add new courses. This function allows them to create course information, including course names, descriptions, and schedules.

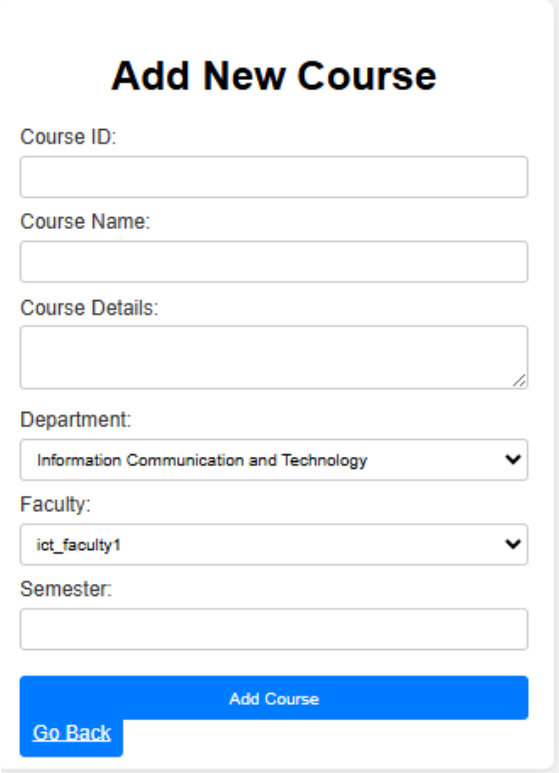


Fig.25: Admin Add Courses Page

26. Admin: Delete Courses

Description: Admins can delete existing courses. This function allows them to remove courses from the system.

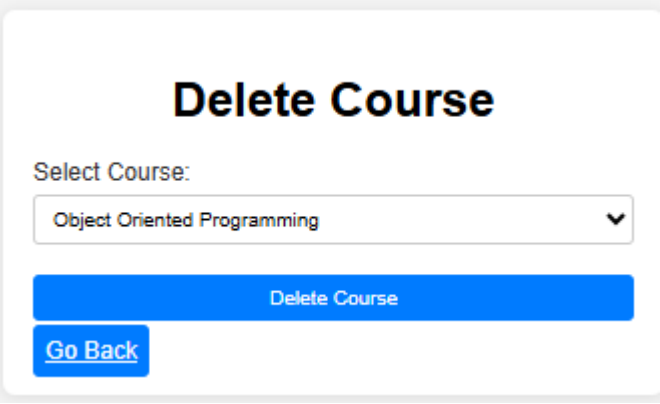


Fig.26: Admin Delete Courses Page

27. Admin: Update Courses

Description: Admins can update the information of existing courses. This function allows them to edit course details as needed.

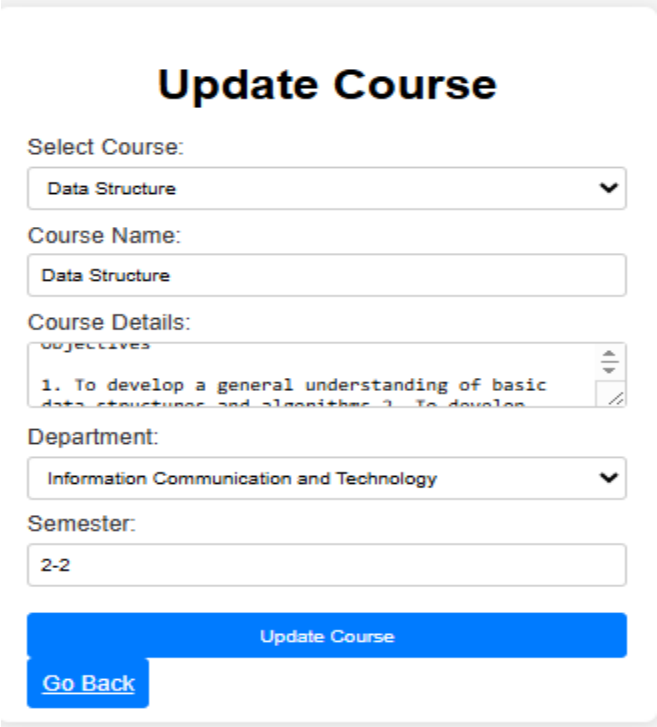


Fig.27: Admin Update Courses

28. Admin: Add Departments

Description: Admins can create new departments. This function allows them to manage department information, including department names and associated courses.

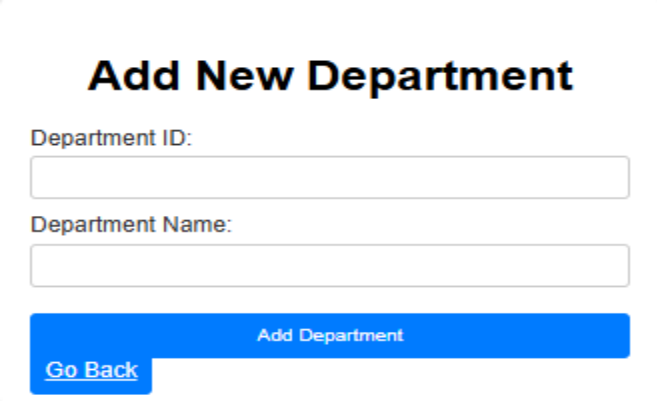


Fig.28: Admin Add Departments Page

29. Admin: Delete Departments

Description: Admins can delete existing departments. This function allows them to remove departments from the system.

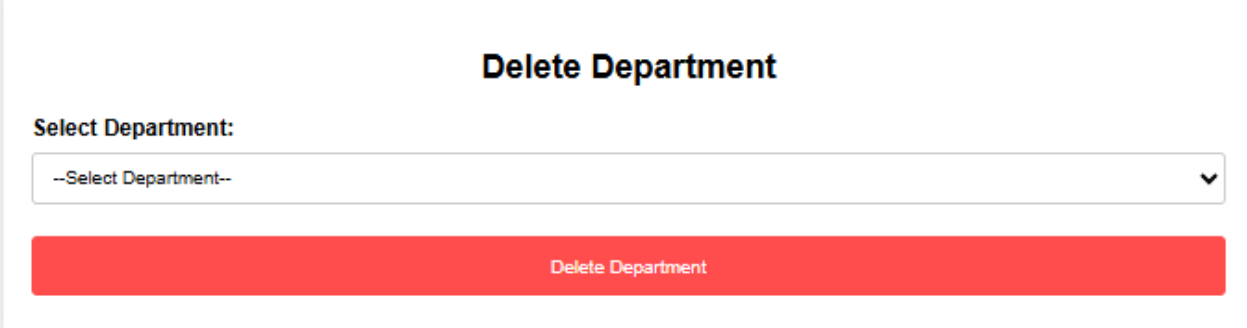


Fig.29: Admin Delete Departments Page

30. Admin: Add Notices

Description: Admins can upload new notices. This function allows them to post important announcements and updates.

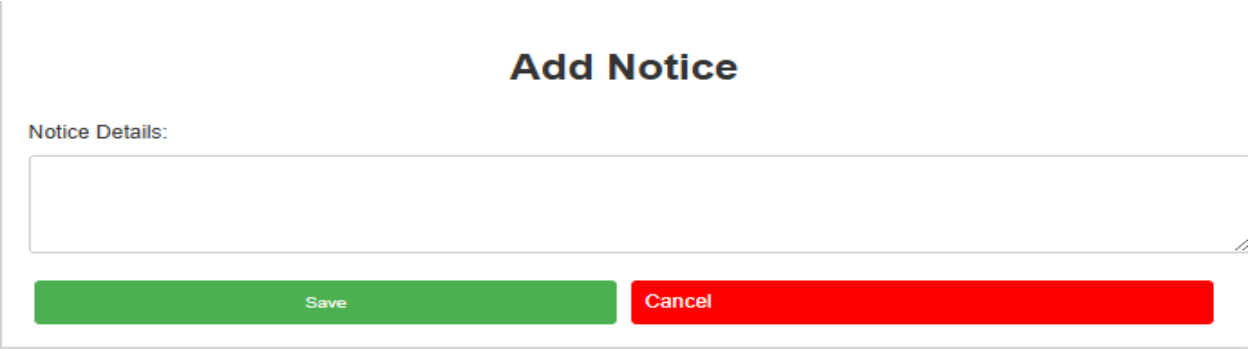


Fig.30: Admin Add Notices Page

31. Admin: Delete Notices

Description: Admins can delete existing notices. This function allows them to remove outdated or irrelevant announcements.

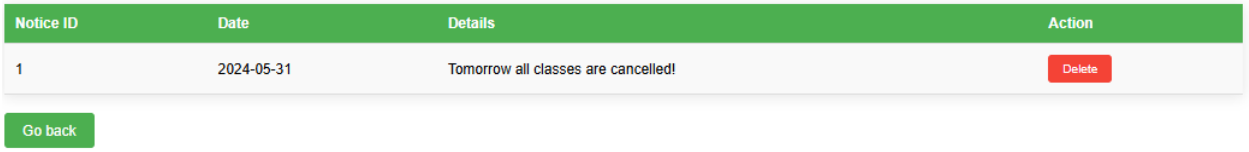


Fig.31: Admin Delete Notices Page

32. Admin: View All Notices

Description: Admins can view all notices. This function allows them to manage and review all posted announcements.

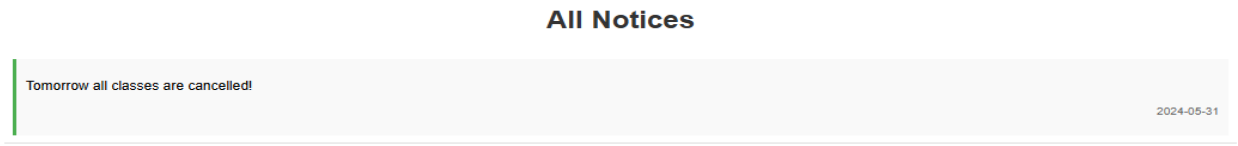


Fig.32: Admin View All Notices Page

33. Admin: Attendance Analysis

Description: Admins can visualize the average attendance of each course based on the available data. This function provides detailed metrics and performance indicators.

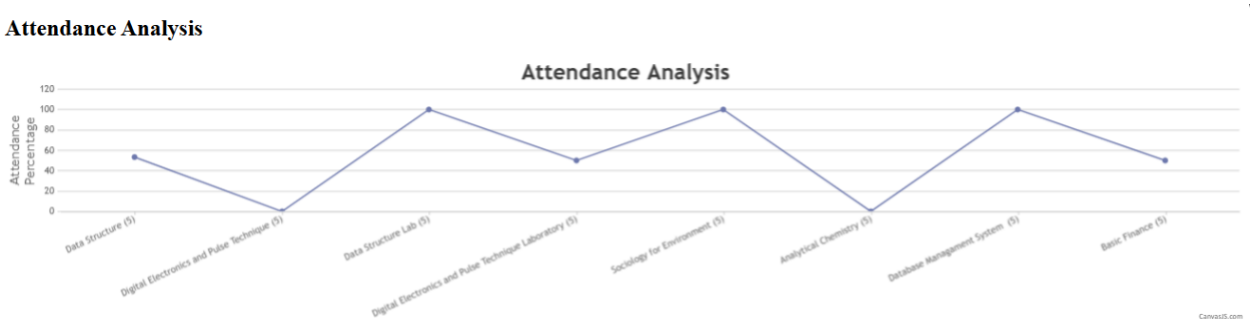


Fig.33: Admin Attendance Analysis Page

34. Admin: View Course Performance

Description: Admins can visualize detailed performance metrics for individual courses, including average marks of each course and name of each course.

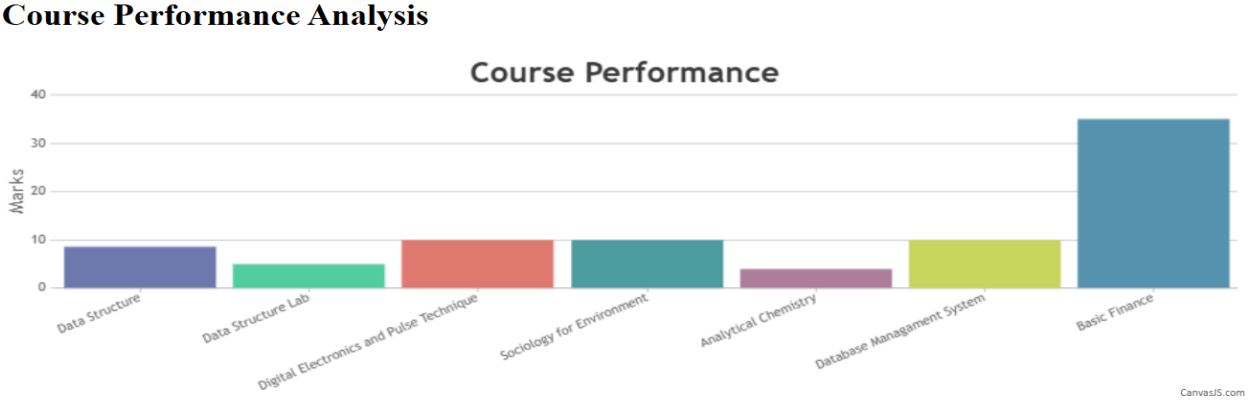


Fig.34: Admin Course Performance Page

35. Admin: User Statistics:

Admins can generate statistics based on the available data. This function provides elaborate metrics and performance indicators in graphs, pie charts etc.

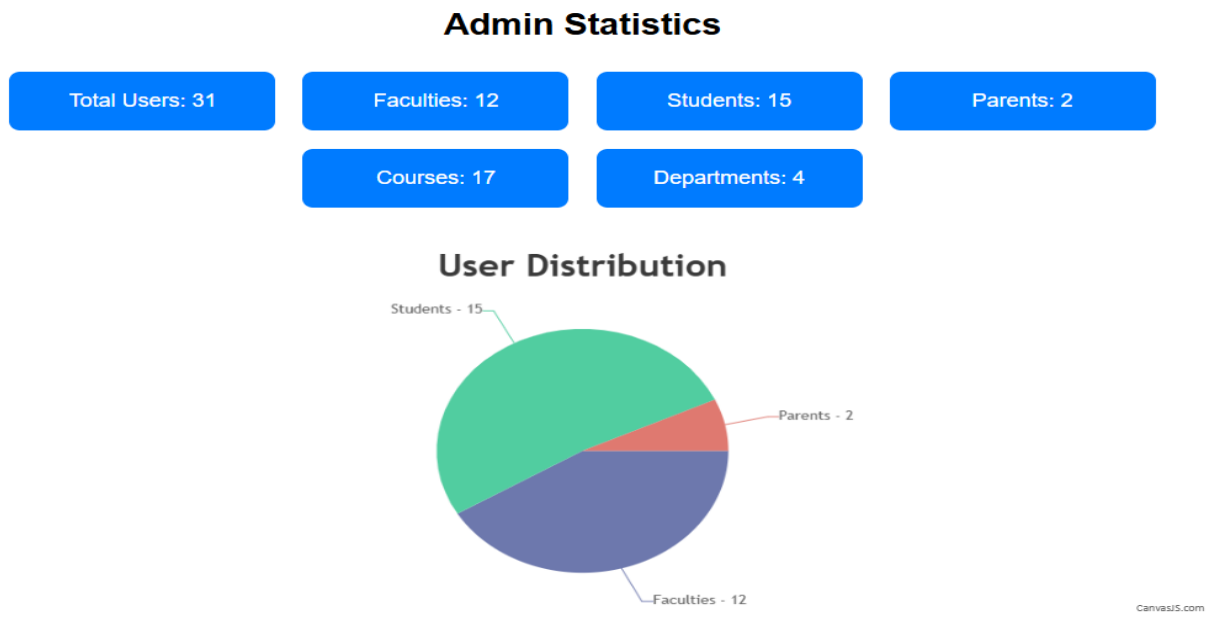


Fig.35.1: Admin User Statistics Page

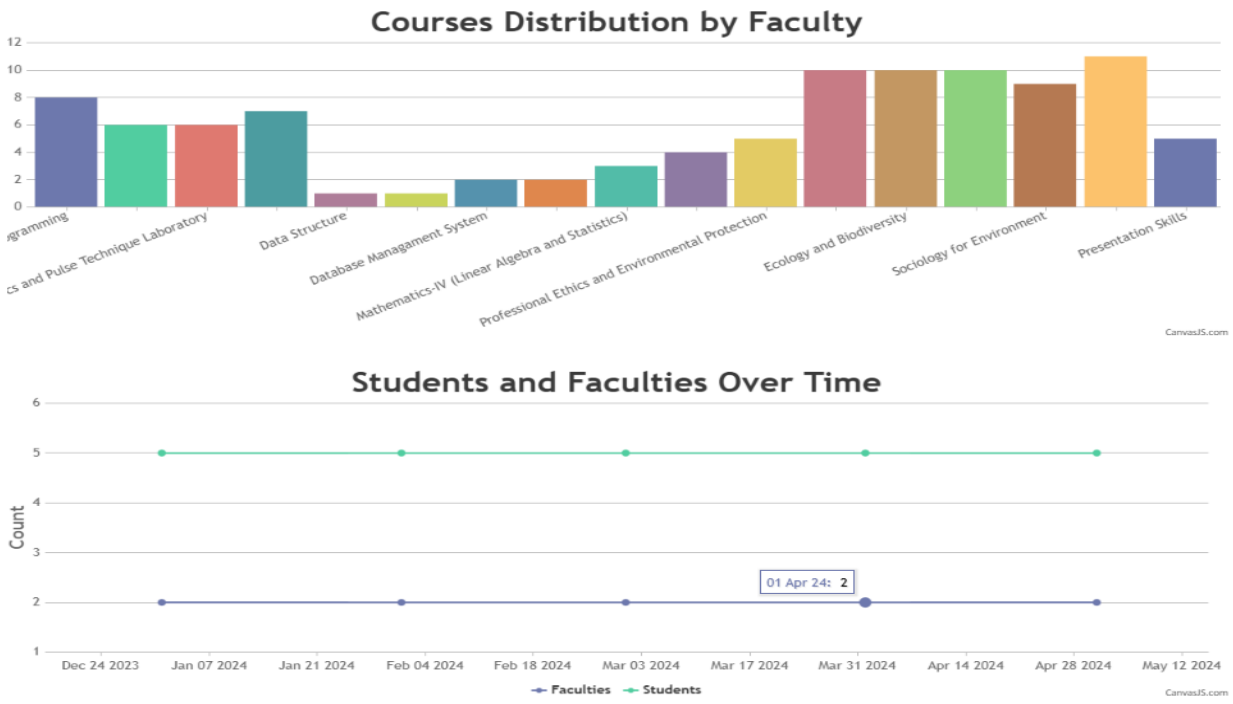
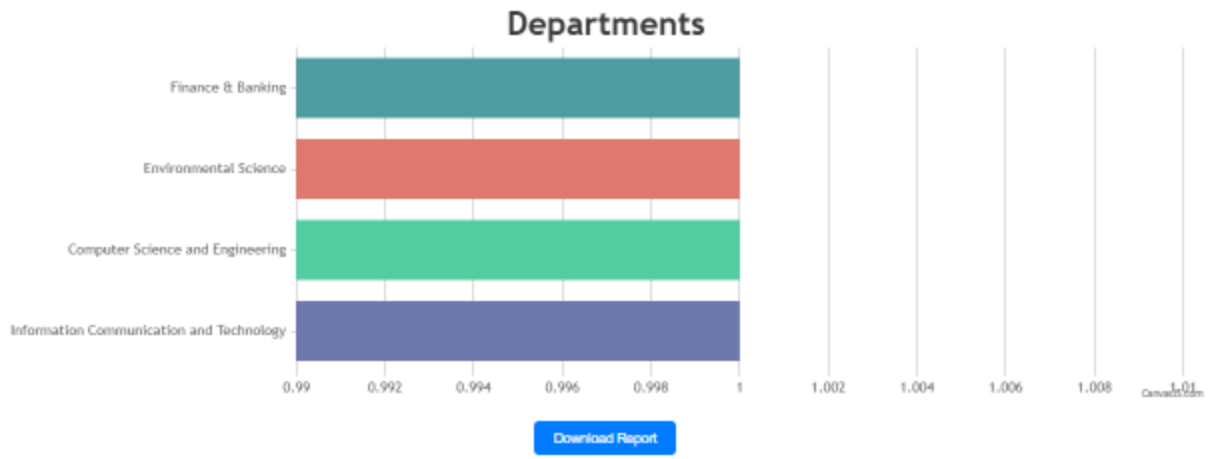


Fig.35.2: Admin User Statistics Page



Courses:

Object Oriented Programming
Digital Electronics and Pulse Technique
Digital Electronics and Pulse Technique Laboratory
Statistics
Data Structure
Data Structure Lab
Database Managament System
Database Managament System Laboratory
Mathematics-IV (Linear Algebra and Statistics)
Discrete Math and Probability
Professional Ethics and Environmental Protection
Environmental Microbiology
Ecology and Biodiversity
Analytical Chemistry
Sociology for Environment
Basic Finance
Presentation Skills

Departments:

Information Communication and Technology
Computer Science and Engineering
Environmental Science
Finance & Banking

Fig.35.3: Admin User Statistics Page

36. Admin: See Patterns/Trends:

Admins can see current trends and patterns based on the available data. This function allows the admin to keep up with the ongoing information of the database.

Patterns and Trends

Attendance Patterns by Day of the Week:

Day of the Week	Average Attendance
6	0.71
5	0.50
2	0.00
7	0.00

Average Marks by Semester:

Semester	Average Marks
1	8.09
2-2	8.00
3-1	35.00

Faculty Activity Patterns:

Faculty ID	Number of Submissions
1	10
6	5
2	5
10	3
9	3
11	1

Fig.36.1:Admin:See Patterns/Trends

Student Attendance Patterns:

Student Name	Average Attendance
Asif	0.60
Joshua	0.50
Lamia	0.33
Maliha	1.00
Nafees	0.60
Nibir	0.33
Niloy	0.50
Omi	0.80
Paromita	0.33
Rafsan	0.33
Ramisa	0.33
Shafin	0.50
Tahsin	0.80
Zawad	0.80

Department Popularity:

Department Name	Student Count
Computer Science and Engineering	6
Information Communication and Technology	5
Environmental Science	3
Finance & Banking	1

Fig.36.2:Admin:See Patterns/Trends

37. Admin: Generate Reports

Description: Admins can generate various reports based on the available data. This function provides insights into the institution's operations.

Admin Report

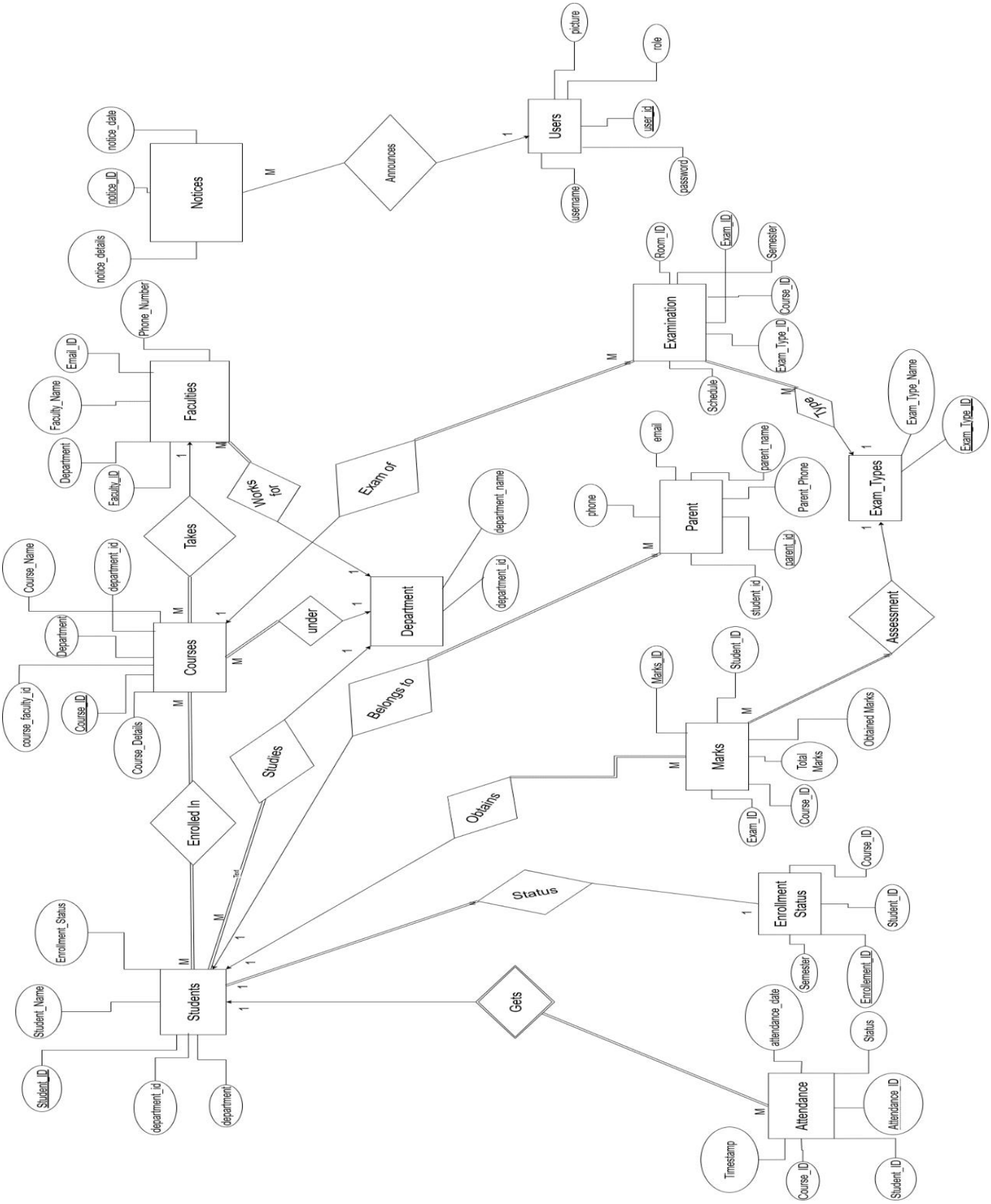
Student ID	Student Name	Enrollment Status	Department
1	Omi	Enrolled	Information Communication and Technology
2	Zawad	Enrolled	Information Communication and Technology
3	Asif	Enrolled	Information Communication and Technology
4	Nafees	Enrolled	Information Communication and Technology
5	Tahsin	Enrolled	Information Communication and Technology
6	Nibir	Enrolled	Computer Science and Engineering
7	Rafsan	Enrolled	Computer Science and Engineering
8	Ramisa	Enrolled	Computer Science and Engineering
9	Lamia	Enrolled	Computer Science and Engineering
10	Paromita	Enrolled	Computer Science and Engineering
11	Joshua	Enrolled	Environmental Science
12	Shafin	Enrolled	Environmental Science
13	Maliha	Enrolled	Environmental Science
14	Niloy	Enrolled	Finance & Banking
15	Imran	Enrolled	Computer Science and Engineering

Faculty Details

Faculty ID	Faculty Name	Department	Email	Phone Number
1	ict_faculty1	Information Communication and Technology	ict_faculty1@gmail.com	1234567890
2	ict_faculty2	Information Communication and Technology	ict_faculty2@gmail.com	1234567890
3	ict_faculty3	Information Communication and Technology	ict_faculty3@gmail.com	1234567890
4	ict_faculty4	Information Communication and Technology	ict_faculty4@yahoo.com	01234567890
5	ict_faculty5	Information Communication	ict_faculty5@hotmail.com	1234567890

Fig.37: Admin Generate Report Page

Entity Relationship Diagram (Updated):



Narrative Description of ER-Diagram

Entities:

Users

Users are the main entity representing all individuals who can access the system. Users provide their user_id (primary key), username (unique), password, role (which can be 'admin', 'student', 'parent', or 'faculty'), and an optional picture. Only the Admin from users table can announce multiple notices. The primary key is user_id, and username is a unique key to ensure no duplicate usernames.

Students

Students are uniquely identified by Student_ID (primary key). Students provide their Student_Name, Enrollment_Status, department, and department_id (foreign key referencing the department table). Students can enroll in multiple courses, get attendance records, and obtain marks. Each student studies in one department, and a department can have many students (1-to-Many relationship).

Faculties

Faculty members are identified by Faculty_ID (primary key) and provide Faculty_Name, Department, Email_ID, and Phone_Number. Faculties can teach multiple courses and work for one department. Each department can have many faculty members (1-to-Many relationship).

Parents

Parents are identified by parent_id (primary key) and provide parent_name, student_id (foreign key linking to the respective student), email, and phone. Parents can assess their child's performance through marks and attendance records.

Courses

Courses have a Course_ID (primary key) and include Course_Name, Course_Details, Department, department_id (foreign key), and course_faculty_id (foreign key referencing the faculties table). Courses are related to multiple students through enrollment and have attendance records, marks, and examinations. Each course belongs to one department, and a department can offer multiple courses (1-to-Many relationship).

Enrollment Status

The Enrollment Status entity records the enrollment of students in courses. It includes Enrollment_ID (primary key), Student_ID (foreign key), Course_ID (foreign key), and Semester. Each student has a status indicating their enrollment in different courses.

Attendance

Attendance records are maintained with Attendance_ID (primary key). It includes Student_ID (foreign key), Course_ID (foreign key), Timestamp, Status (Present/Absent), and attendance_date. Students get attendance records for courses. Faculties can record and update attendance.

Marks

Marks obtained by students are recorded in the Marks table, with Marks_ID (primary key). It includes Exam_ID (foreign key), Course_ID (foreign key), Total_Marks, Obtained_Marks, and Student_ID (foreign key). Marks belong to students and courses, and students obtain marks through assessments.

Exam Types

Exam Types define the types of exams conducted, with Exam_Type_ID (primary key) and Exam_Type_Name as an attribute. Each examination is associated with a specific exam type.

Examinations

The Examinations table records exam schedules and details. It includes Exam_ID (primary key), Semester, Course_ID (foreign key), Schedule, Exam_Type_ID (foreign key), and Room_ID. Examinations are related to courses and have specific types.

Notices

Notices are announcements made by the admin, recorded with notice_ID (primary key). Each notice has a notice_date and notice_details. Notices are announced by users.

Relationships:

Enrolled In: A student can enroll in multiple courses, and a course can have multiple students enrolled.

Takes: A faculty member can teach multiple courses, but each course is assigned to one faculty member.

Belongs to: Courses belong to a department, and each department can offer multiple courses.

Gets: Attendance records are linked to students and courses.

Obtains: Students receive marks for courses through examinations.

Assessment: Parents can assess their child's performance through marks.

Exam of: Examinations are related to courses and have specific types.

Announces: Admin users can announce multiple notices, and each notice is announced by an admin.

Status: Students have a status relationship with their enrollment in different courses, indicating their enrollment status.

Type: Examinations have a type relationship with exam types, defining the specific type of each examination.

Studies: Each student is associated with one department (Total Participation) and each department can have multiple students (Partial Participation).

Works for: Each faculty works for one department (Total Participation) and each department can have multiple faculties (Partial Participation).

Progress Report:

Summary

- Selection of Topics

- User Wise Functionalities Discussion

- Tables and Attributes add

- Key Constraints and Participation Constraints Add

- ER Diagram Design

- Finalizing ER Diagram and Understanding Integrity Constraints

- Writing description of Diagram

- Learning Relational Schema for next phase

Group Progress Report: **Reporting Period: 28.04.2021**

Team Member	Total Hours (period)	Remarks
Prithviraj Chowdhury (2254901101)	25.03.2024 08:00PM-10:30PM (02 hours 30 mins)	Project Planning, Initial Description of ER Diagram, Database Schema Overview Introduction of the Project, Database Relations Discussion, Attribute and Key Constraints Finalizing ER Diagram Design, Review and Edits, Document Formatting
Gazi Mohammad Abrar Zawad (2254901049)	11.04.2024 06:00PM-08:45PM (02 hours 45 mins)	Writing Descriptions for ER Diagram, Writing Functional Descriptions, Group Meeting Notes Group Discussion Notes and Decision Making ER Diagram Creation, Detailing Relationships, and Keys, Group Discussion
Jablay Noor Rahman (2254901093)	27.04.2024 07:30PM-10:00PM (03 hours 30 mins)	Reviewing ER Diagram Concepts, Attribute Analysis, Initial Report Drafting Group Discussion Notes and Overall Outcome of All Group Meetings Submission preparation of ER Diagram design, Document Formatting
Group Totals	8 Hours 45 mins	

Individual Report:

Member Name: Prithviraj Chowdhury Reporting Period: 28.04.2024

Date	Hours	Activities
10.04.2024	2 hours 42 minutes	Implementing Relational Database concepts on MySQL
15.04.2024	3 hours	Solved problems regarding ER diagram from textbooks
17.04.2024	2 hours	Revised class lectures regarding attributes, cardinalities, participation constraints
05.04.2024 18.04.2024 25.04.2025	8 hours 20 minutes	Project Initialization Group meetings
19.04.2024	1 hour 30 minutes	Attempted demo work through YouTube tutorials and project samples, determining key constraints
22.04.2024	1 hour	Learned basic HTML and CSS from online tutorials
27.04.2024	1 hour 30 minutes	Learning Relational Schema for Project Phase-2

Member Name: Gazi Mohammad Abrar Zawad Reporting Period: 28.04.2024

Date	Hours	Activities
11.04.2024	2 hours	Learning Entity Relationship Diagram from class notes and YouTube videos
18.04.2024	2 hours	Watched videos on ER diagram concepts, worked on exercises
19.04.2024	2 hours	Revising class notes regarding Tables, Attributes, Degree of Relationship Set, Key Constraints
05.04.2024 18.04.2024 25.04.2024	5 hours 20 mins	Group meetings
23.04.2024	1 hour 30 mins	Writing introduction, description of ER Diagram
25.04.2024	30 mins	Learned basic HTML and CSS from online tutorials
27.04.2024	1 hour 30 minutes	Reviewing the entire project functionalities and project report

Member Name: Jablay Noor Rahman Reporting Period: 28.04.2024

Date	Hours	Activities
16.04.2024	1 hour 17 minutes	Watched ER diagram lectures
23.04.2024	1 hour 30 minutes	Worked on exercises based on ER Model Concept
13.04.2024	45 minutes	Studied textbook on Database Management System
05.04.2024 18.04.2024 25.04.2024	5 hours 20 mins	Group meetings
30.04.2024	3 hours	Learned basic HTML and CSS from online tutorials
31.04.2024	1 hour	Worked on exercises regarding Keys in RDBMS
27.04.2024	1 hour 30 minutes	Hardcopy of ER Diagram design, Decided the number of tables and relationships

PHASE-02

1. Schema refinement-

The ER diagram submitted in Phase-1 has been modified and converted into relational schema which is given below:

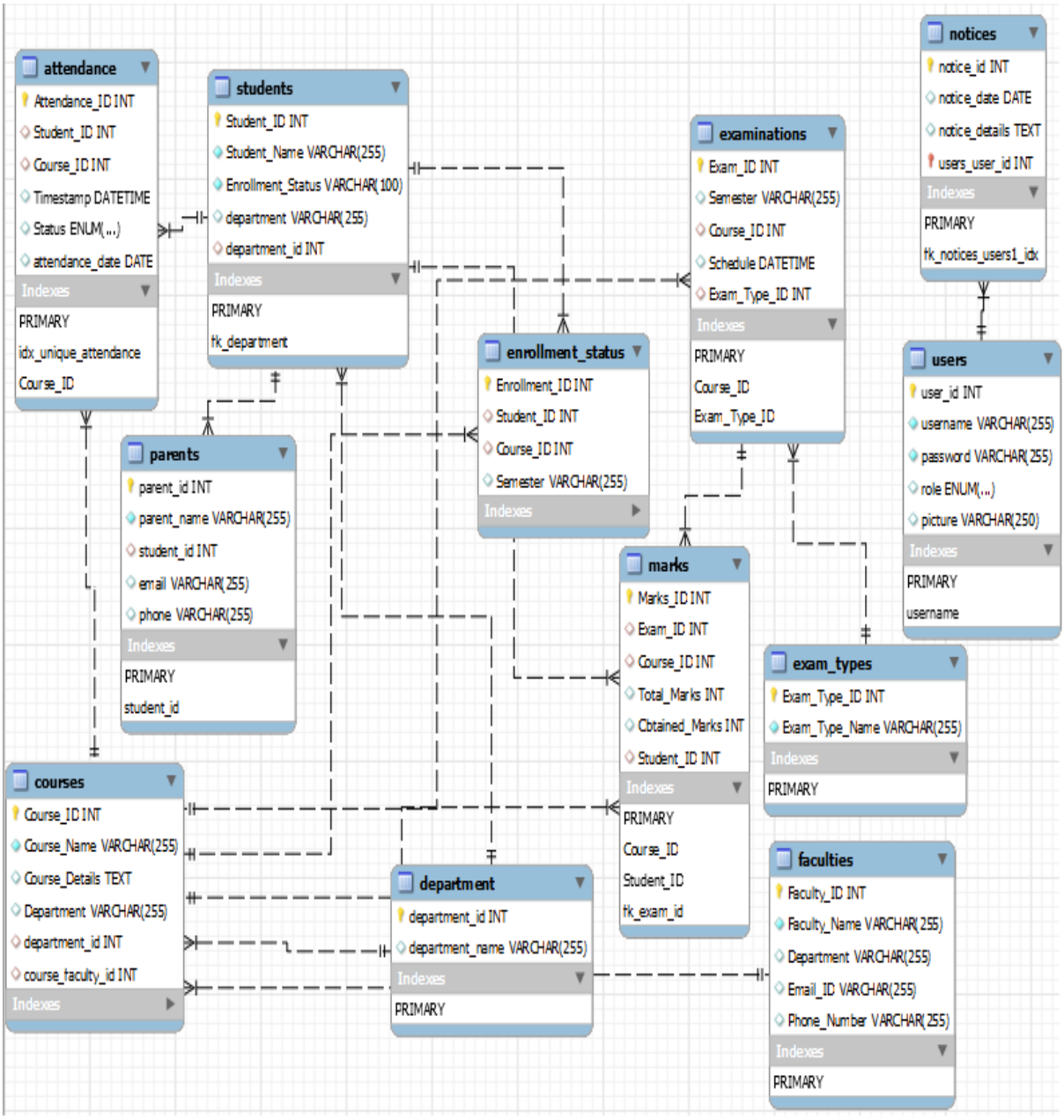


Figure.1: Relational Schema

Integrity Constraints:

Attendance_ID(PK)	Student_ID(FK)	Course_ID(FK)	Timestamp	Status	Attendance_date

Table.1: Attendance Table

Course_ID(PK)	Course_Name	Course_Details	Department	department_id(FK)	course_faculty_id(FK)

Table.2: Courses Table

department_id (PK)	department_name

Table.3: Department Table

Enrollment_ID(PK)	Student_ID(FK)	Course_ID(FK)	Semester

Table.4: Enrollment_Status Table

Exam_Type_ID (PK)	Exam_Type_Name

Table.5: Exam_Types Table

Exam_ID (PK)	Semester	Course_ID(FK)	Schedule	Exam_Type_ID(FK)

Table6: Examinations Table

Faculty_ID(PK)	Faculty_Name	Department	Email_ID	Phone_Number

Table.7: Faculties Table

Marks_ID(PK)	Exam_ID(FK)	Course_ID(FK)	Total_Marks	Obtained_Marks	Student_ID(FK)

Table.8: Marks Table

Student_ID(PK)	Student_Name	Enrollement_Status	department	deparment_id

Table.9: Students Table

parent_id (PK)	parent_name	student_id	email	phone

Table.10: Parents Table

Student_ID (PK)	Student_Name	Enrollement_Status	department	deparment_id

Table.9: Students Table

notice_id (PK)	notice_date	notice_details

Table.11: Notices Table

user_id (PK)	username	password	role	picture

Table.12: Users Table

2. Normalization:

1st Normal Form (1NF):

The given relation is in 1st Normal Form because it meets the following criteria:

1. All cells contain atomic (indivisible) values.

2. Entries in each column are of the same type.

3. Each tuple is unique; there are no duplicate rows.

4. Column names are unique.

2nd Normal Form (2NF):

The given relation is in 2nd Normal Form because it satisfies these conditions:

1. The table is already in 1st Normal Form.

2. There is no partial dependency; no proper subset of a candidate key can uniquely identify a non-prime attribute.

3. All non-key attributes are fully dependent on the entire primary key.

3rd Normal Form (3NF):

The given relation is in 3rd Normal Form because it meets the following requirements:

1. The table is in 2nd Normal Form.

2. There is no transitive dependency; no non-prime attribute can determine another non-prime attribute.

3. For each functional dependency, either the left-hand side is a super key, or the right-hand side is a non-prime attribute.

4. All columns are determined only by the primary key and no other column.

1. Attendance Table:

```
mysql> desc attendance;
+-----------------+--------------------------+------+-----+-------------------+-------------------+
| Field           | Type                     | Null | Key | Default           | Extra             |
+-----------------+--------------------------+------+-----+-------------------+-------------------+
| Attendance_ID   | int                      | NO   | PRI | NULL              | auto_increment    |
| Student_ID      | int                      | YES  | MUL | NULL              |                   |
| Course_ID       | int                      | YES  | MUL | NULL              |                   |
| Timestamp       | datetime                 | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| Status          | enum('Present','Absent') | YES  |     | NULL              |                   |
| attendance_date | date                     | YES  |     | NULL              |                   |
+-----------------+--------------------------+------+-----+-------------------+-------------------+
6 rows in set (0.01 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Attendance_ID`, and all non-key attributes (Student_ID, Course_ID, Timestamp, Status, attendance_date) are fully dependent on it.

Third Normal Form	☑	There are no transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Attendance table is in 3rd Normal Form or 3NF.

2. Courses Table:

```
mysql> desc courses;
+--------------------+--------------+------+-----+---------+----------------+
| Field              | Type         | Null | Key | Default | Extra          |
+--------------------+--------------+------+-----+---------+----------------+
| Course_ID          | int          | NO   | PRI | NULL    | auto_increment |
| Course_Name        | varchar(255) | NO   |     | NULL    |                |
| Course_Details     | text         | YES  |     | NULL    |                |
| Department         | varchar(255) | YES  |     | NULL    |                |
| department_id      | int          | YES  | MUL | NULL    |                |
| course_faculty_id  | int          | YES  | MUL | NULL    |                |
+--------------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Course_ID`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Courses table is in 3rd Normal Form or 3NF.

3. Department Table:

```
mysql> desc department;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| department_id   | int          | NO   | PRI | NULL    |       |
| department_name | varchar(255) | YES  |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `department_id`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Department table is in 3rd Normal Form or 3NF.

4. Enrollment_Status Table:

```
mysql> desc enrollment_status;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| Enrollment_ID | int          | NO   | PRI | NULL    | auto_increment |
| Student_ID    | int          | YES  | MUL | NULL    |                |
| Course_ID     | int          | YES  | MUL | NULL    |                |
| Semester      | varchar(255) | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Enrollment_ID`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Enrollment_Status table is in 3rd Normal Form or 3NF.

5. Exam_Types Table:

```
mysql> desc exam_types;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| Exam_Type_ID   | int          | NO   | PRI | NULL    | auto_increment |
| Exam_Type_Name | varchar(255) | NO   |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Exam_Type_ID`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Exam_Types table is in 3rd Normal Form or 3NF.

6. Examinations Table:

```
mysql> desc examinations;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| Exam_ID      | int          | NO   | PRI | NULL    | auto_increment |
| Semester     | varchar(255) | YES  |     | NULL    |                |
| Course_ID    | int          | YES  | MUL | NULL    |                |
| Schedule     | datetime     | YES  |     | NULL    |                |
| Exam_Type_ID | int          | YES  | MUL | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql>
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Exam_ID`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Examinations table is in 3rd Normal Form or 3NF.

7. Faculties Table:

```
mysql> desc faculties;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| Faculty_ID   | int          | NO   | PRI | NULL    | auto_increment |
| Faculty_Name | varchar(255) | NO   |     | NULL    |                |
| Department   | varchar(255) | YES  |     | NULL    |                |
| Email_ID     | varchar(255) | YES  |     | NULL    |                |
| Phone_Number | varchar(255) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Faculty_ID`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Faculties table is in 3rd Normal Form or 3NF.

8. Marks Table:

```
mysql> desc marks;
+----------------+------+------+-----+---------+----------------+
| Field          | Type | Null | Key | Default | Extra          |
+----------------+------+------+-----+---------+----------------+
| Marks_ID       | int  | NO   | PRI | NULL    | auto_increment |
| Exam_ID        | int  | YES  | MUL | NULL    |                |
| Course_ID      | int  | YES  | MUL | NULL    |                |
| Total_Marks    | int  | YES  |     | NULL    |                |
| Obtained_Marks | int  | YES  |     | NULL    |                |
| Student_ID     | int  | YES  | MUL | NULL    |                |
+----------------+------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Marks_ID`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Marks table is in 3rd Normal Form or 3NF.

9. Notices Table:

```
mysql> desc notices;
+----------------+------+------+-----+---------+----------------+
| Field          | Type | Null | Key | Default | Extra          |
+----------------+------+------+-----+---------+----------------+
| notice_id      | int  | NO   | PRI | NULL    | auto_increment |
| notice_date    | date | YES  |     | NULL    |                |
| notice_details | text | YES  |     | NULL    |                |
+----------------+------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `notice_id`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Notices table is in 3rd Normal Form or 3NF.

10. Parents Table:

```
mysql> desc parents;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| parent_id   | int          | NO   | PRI | NULL    | auto_increment |
| parent_name | varchar(255) | NO   |     | NULL    |                |
| student_id  | int          | YES  | MUL | NULL    |                |
| email       | varchar(255) | YES  |     | NULL    |                |
| phone       | varchar(255) | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `parent_id`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Parents table is in 3rd Normal Form or 3NF.

11. Students Table:

```
mysql> desc students;
+-------------------+--------------+------+-----+---------+----------------+
| Field             | Type         | Null | Key | Default | Extra          |
+-------------------+--------------+------+-----+---------+----------------+
| Student_ID        | int          | NO   | PRI | NULL    | auto_increment |
| Student_Name      | varchar(255) | NO   |     | NULL    |                |
| Enrollment_Status | varchar(100) | NO   |     | NULL    |                |
| department        | varchar(255) | YES  |     | NULL    |                |
| department_id     | int          | YES  | MUL | NULL    |                |
+-------------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `Student_ID`. All non-key attributes are fully dependent on it.

Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Students table is in 3rd Normal Form or 3NF.

12. Users Table:

```
mysql> desc users;
+----------+-----------------------------------------+------+-----+---------+----------------+
| Field    | Type                                    | Null | Key | Default | Extra          |
+----------+-----------------------------------------+------+-----+---------+----------------+
| user_id  | int                                     | NO   | PRI | NULL    | auto_increment |
| username | varchar(255)                            | NO   | UNI | NULL    |                |
| password | varchar(255)                            | NO   |     | NULL    |                |
| role     | enum('admin','student','parent','faculty') | YES  |     | NULL    |                |
| picture  | varchar(250)                            | YES  |     | NULL    |                |
+----------+-----------------------------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Normalization Check:

First Normal Form	☑	All fields contain atomic values. No repeating groups.
Second Normal Form	☑	The primary key is `user_id`. All non-key attributes are fully dependent on it.
Third Normal Form	☑	No transitive dependencies. Each non-key attribute is directly dependent on the primary key.

The Users table is in 3rd Normal Form or 3NF.

3.SQL Statements:

The necessary SQL commands required to create the database ‘dbms_project’ with appropriate foreign keys are shown below:

Creation of ‘dbms_project’ database:

```
create database dbms_project;

use dbms_project;
```

Creation of Users Table:

```
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    role ENUM('admin', 'student', 'parent', 'faculty'),
    picture VARCHAR(250)
);
```

Creation of Department Table:

```
-- Create department table
CREATE TABLE department (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(255)
);
```

Creation of Students Table:

```
CREATE TABLE students (
    Student_ID INT AUTO_INCREMENT PRIMARY KEY,
    Student_Name VARCHAR(255) NOT NULL,
    Enrollment_Status VARCHAR(100) NOT NULL,
    department VARCHAR(255),
    department_id INT,
    CONSTRAINT fk_department FOREIGN KEY (department_id) REFERENCES department(department_id)
);
```

Creation of Parents Table:

```
CREATE TABLE parents (
    parent_id INT AUTO_INCREMENT PRIMARY KEY,
    parent_name VARCHAR(255) NOT NULL,
    student_id INT,
    email VARCHAR(255),
    phone VARCHAR(255),
    CONSTRAINT parents_ibfk_1 FOREIGN KEY (student_id) REFERENCES students(Student_ID)
);
```

Creation of Notices Table:

```
CREATE TABLE notices (
    notice_id INT AUTO_INCREMENT PRIMARY KEY,
    notice_date DATE,
    notice_details TEXT
);
```

Creation of Courses Table:

```
CREATE TABLE courses (
    Course_ID INT AUTO_INCREMENT PRIMARY KEY,
    Course_Name VARCHAR(255) NOT NULL,
    Course_Details TEXT,
    Department VARCHAR(255),
    department_id INT,
    course_faculty_id INT,
```

CONSTRAINT courses_ibfk_1 FOREIGN KEY (department_id) REFERENCES department(department_id),

CONSTRAINT fk_course_department FOREIGN KEY (course_faculty_id) REFERENCES faculties(Faculty_ID)

);

Creation of Enrollment_Status Table:

```
CREATE TABLE enrollment_status (

   Enrollment_ID INT AUTO_INCREMENT PRIMARY KEY,

   Student_ID INT,

   Course_ID INT,

   Semester VARCHAR(255),

   CONSTRAINT enrollment_status_ibfk_1 FOREIGN KEY (Student_ID) REFERENCES students(Student_ID),

   CONSTRAINT enrollment_status_ibfk_2 FOREIGN KEY (Course_ID) REFERENCES courses(Course_ID)

);
```

Creation of Faculties Table:

```
CREATE TABLE faculties (

   Faculty_ID INT AUTO_INCREMENT PRIMARY KEY,

   Faculty_Name VARCHAR(255) NOT NULL,

   Department VARCHAR(255),

   Email_ID VARCHAR(255),

   Phone_Number VARCHAR(255)

);
```

Creation of Examinations Table:

```
CREATE TABLE examinations (
    Exam_ID INT AUTO_INCREMENT PRIMARY KEY,
    Semester VARCHAR(255),
    Course_ID INT,
    Schedule DATETIME,
    Exam_Type_ID INT,
    CONSTRAINT examinations_ibfk_1 FOREIGN KEY (Course_ID) REFERENCES courses(Course_ID),
    CONSTRAINT examinations_ibfk_2 FOREIGN KEY (Exam_Type_ID) REFERENCES exam_types(Exam_Type_ID)
);
```

Creation of Attendance Table:

```
CREATE TABLE attendance (
    Attendance_ID INT AUTO_INCREMENT PRIMARY KEY,
    Student_ID INT,
    Course_ID INT,
    Timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
    Status ENUM('Present', 'Absent'),
    attendance_date DATE,
    CONSTRAINT attendance_ibfk_1 FOREIGN KEY (Student_ID) REFERENCES students(Student_ID),
    CONSTRAINT attendance_ibfk_2 FOREIGN KEY (Course_ID) REFERENCES courses(Course_ID)
);
```

Creation of Exam_Types table:

```
CREATE TABLE exam_types (
    Exam_Type_ID INT AUTO_INCREMENT PRIMARY KEY,
    Exam_Type_Name VARCHAR(255) NOT NULL
);
```

Creation of Marks Table:

```
  Marks_ID INT AUTO_INCREMENT PRIMARY KEY,
    Exam_ID INT,
    Course_ID INT,
    Total_Marks INT,
    Obtained_Marks INT,
    Student_ID INT,
    CONSTRAINT fk_exam_id FOREIGN KEY (Exam_ID) REFERENCES examinations(Exam_ID),
    CONSTRAINT marks_ibfk_2 FOREIGN KEY (Course_ID) REFERENCES courses(Course_ID),
    CONSTRAINT marks_ibfk_3 FOREIGN KEY (Student_ID) REFERENCES students(Student_ID)
);
```

4.Populating the Database:

Set of data have been inserted into each relation of the created database 'dbms_project'. After insertion of data, the tables are currently in this form:

Attendance Table:

```
mysql> select * from attendance;
```

Attendance_ID	Student_ID	Course_ID	Timestamp	Status	attendance_date
1	1	2201	2024-05-31 17:03:56	Present	2024-05-31
2	2	2201	2024-05-31 17:03:56	Present	2024-05-31
3	3	2201	2024-05-31 17:03:56	Absent	2024-05-31
4	4	2201	2024-05-31 17:03:56	Absent	2024-05-31
5	5	2201	2024-05-31 17:03:56	Present	2024-05-31
6	6	1109	2024-05-31 17:25:31	Absent	2024-05-31
7	7	1109	2024-05-31 17:25:31	Absent	2024-05-31
8	8	1109	2024-05-31 17:25:31	Absent	2024-05-31
9	9	1109	2024-05-31 17:25:31	Absent	2024-05-31
10	10	1109	2024-05-31 17:25:31	Absent	2024-05-31
11	1	2202	2024-05-31 18:11:46	Present	2024-05-31
12	2	2202	2024-05-31 18:11:46	Present	2024-05-31
13	3	2202	2024-05-31 18:11:46	Present	2024-05-31
14	4	2202	2024-05-31 18:11:46	Present	2024-05-31
15	5	2202	2024-05-31 18:11:46	Present	2024-05-31
16	1	2201	2024-05-31 18:31:57	Present	2024-05-30
17	2	2201	2024-05-31 18:31:57	Present	2024-05-30
18	3	2201	2024-05-31 18:31:57	Present	2024-05-30
19	4	2201	2024-05-31 18:31:57	Present	2024-05-30
20	5	2201	2024-05-31 18:31:57	Present	2024-05-30
21	1	2201	2024-05-31 18:32:06	Absent	2024-05-20
22	2	2201	2024-05-31 18:32:06	Absent	2024-05-20
23	3	2201	2024-05-31 18:32:06	Absent	2024-05-20
24	4	2201	2024-05-31 18:32:06	Absent	2024-05-20

Courses Table:

```
mysql> select * from courses;
```

Course_ID	Course_Name	Course_Details	Department	department_id	course_faculty_id
1108	Object Oriented Programming	To learn the concept of OOP with a pure object-oriented programming language (Java) To learn how to use advance programming features such as GUI design, exception handling and multithreading. To learn how to design and develop a complete real-world software solution.	Computer Science and Engineering	2	8
1109	Digital Electronics and Pulse Technique	To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronic circuits. To prepare students to perform the analysis and design of various combinational and sequential circuits using gates	Computer Science and Engineering	2	6
1110	Digital Electronics and Pulse Technique Laboratory	To acquaint the students with the fundamental concepts in classical manual digital design To familiarize the students clearly with the way in which digital circuits are designed today using CAD tools like Schematic Capture and Verilog HDL. To develop students? analytical skills to build complex digital circuit and impart the knowledge about ?Green Technology? to integrate it in their projects			

Department Table:

```
mysql> select * from department;
+---------------+-----------------------------------------+
| department_id | department_name                         |
+---------------+-----------------------------------------+
|             1 | Information Communication and Technology |
|             2 | Computer Science and Engineering        |
|             3 | Environmental Science                   |
|          9909 | Finance & Banking                       |
+---------------+-----------------------------------------+
4 rows in set (0.01 sec)
```

Enrollment_Status Table:

```
mysql> select * from enrollment_status;
+---------------+------------+-----------+----------+
| Enrollment_ID | Student_ID | Course_ID | Semester |
+---------------+------------+-----------+----------+
|             1 |          1 |      2201 | 2-2      |
|             2 |          1 |      2202 | 2-2      |
|             3 |          1 |      2203 | 2-2      |
|             4 |          1 |      2204 | 2-2      |
|             5 |          1 |      2207 | 2-2      |
|             6 |          1 |      2209 | 2-2      |
|             7 |          1 |      2211 | 2-2      |
|             8 |          2 |      2201 | 1        |
|             9 |          2 |      2202 | 1        |
|            10 |          2 |      2203 | 1        |
|            11 |          2 |      2204 | 1        |
|            12 |          2 |      2207 | 1        |
|            13 |          2 |      2209 | 1        |
|            14 |          2 |      2211 | 1        |
|            15 |          3 |      2201 | 1        |
|            16 |          3 |      2202 | 1        |
|            17 |          3 |      2203 | 1        |
|            18 |          3 |      2204 | 1        |
|            19 |          3 |      2207 | 1        |
|            20 |          3 |      2209 | 1        |
|            21 |          3 |      2211 | 1        |
|            22 |          4 |      2201 | 1        |
|            23 |          4 |      2202 | 1        |
|            24 |          4 |      2203 | 1        |
|            25 |          4 |      2204 | 1        |
|            26 |          4 |      2207 | 1        |
|            27 |          4 |      2209 | 1        |
|            28 |          4 |      2211 | 1        |
|            29 |          5 |      2201 | 1        |
|            30 |          5 |      2202 | 1        |
```

Exam_Types table:

```
mysql> select * from exam_types;
+--------------+-------------------------------------+
| Exam_Type_ID | Exam_Type_Name                      |
+--------------+-------------------------------------+
|            1 | Class test-1                        |
|            2 | Class test-2                        |
|            3 | Class test-3                        |
|            4 | Class test-4                        |
|            5 | Assignment/Presentation/Term paper  |
|            6 | Mid Term                            |
|            7 | Attendance                          |
+--------------+-------------------------------------+
7 rows in set (0.00 sec)
```

Examinations Table:

```
mysql> select * from examinations;
+---------+----------+-----------+---------------------+--------------+
| Exam_ID | Semester | Course_ID | Schedule            | Exam_Type_ID |
+---------+----------+-----------+---------------------+--------------+
|       1 | 2-2      |      2201 | 2024-05-31 14:03:20 |            3 |
|       2 | 2-2      |      2202 | 2024-05-31 14:32:44 |            5 |
|       3 | 2-2      |      1109 | 2024-05-31 14:35:07 |            3 |
|       4 | 2-2      |      3306 | 2024-05-31 14:47:33 |            4 |
|       5 | 2-2      |      3304 | 2024-05-31 14:48:12 |            1 |
|       6 | 2-2      |      2203 | 2024-05-31 14:50:38 |            1 |
|       7 | 3-1      |      8909 | 2024-05-31 16:30:05 |            6 |
+---------+----------+-----------+---------------------+--------------+
7 rows in set (0.00 sec)

mysql>
```

Faculties Tables:

```
mysql> select * from faculties;
+------------+-------------+-----------------------------------------+-------------------------+--------------+
| Faculty_ID | Faculty_Name | Department                              | Email_ID                | Phone_Number |
+------------+-------------+-----------------------------------------+-------------------------+--------------+
|          1 | ict_faculty1 | Information Communication and Technology | ict_faculty1@gmail.com  | 1234567890   |
|          2 | ict_faculty2 | Information Communication and Technology | ict_faculty2@gmail.com  | 1234567890   |
|          3 | ict_faculty3 | Information Communication and Technology | ict_faculty3@gmail.com  | 1234567890   |
|          4 | ict_faculty4 | Information Communication and Technology | ict_faculty4@yahoo.com  | 01234567890  |
|          5 | ict_faculty5 | Information Communication and Technology | ict_faculty5@hotmail.com | 1234567890   |
|          6 | cse_faculty1 | Computer Science and Engineering        | cse@gmail.com           | 1234567890   |
|          7 | cse_faculty2 | Computer Science and Engineering        | cse@gmail.com           | 1234567890   |
|          8 | cse_faculty3 | Computer Science and Engineering        | cse@gmail.com           | 1234567890   |
|          9 | es_faculty1  | Environmental Science                   | es@gmail.com            | 1234567890   |
|         10 | es_faculty2  | Environmental Science                   | es@gmail.com            | 1234567890   |
|         11 | fin_faculty  | Finance & Banking                       | fin@gmail.com           | 1234567890   |
|         12 | es_faculty3  | Information Communication and Technology | es@gmail.com            | 1234567890   |
+------------+-------------+-----------------------------------------+-------------------------+--------------+
```

Marks Table:

```
mysql> select * from marks;
+----------+---------+-----------+-------------+----------------+------------+
| Marks_ID | Exam_ID | Course_ID | Total_Marks | Obtained_Marks | Student_ID |
+----------+---------+-----------+-------------+----------------+------------+
|        1 |       1 |      2201 |          10 |              9 |          1 |
|        2 |       1 |      2201 |          10 |              8 |          2 |
|        3 |       1 |      2201 |          10 |              7 |          3 |
|        4 |       1 |      2201 |          10 |             10 |          4 |
|        5 |       1 |      2201 |          10 |              9 |          5 |
|        6 |       2 |      2202 |          10 |              5 |          1 |
|        7 |       2 |      2202 |          10 |              5 |          2 |
|        8 |       2 |      2202 |          10 |              5 |          3 |
|        9 |       2 |      2202 |          10 |              5 |          4 |
|       10 |       2 |      2202 |          10 |              5 |          5 |
|       11 |       3 |      1109 |          10 |             10 |          6 |
|       12 |       3 |      1109 |          10 |             10 |          7 |
|       13 |       3 |      1109 |          10 |             10 |          8 |
|       14 |       3 |      1109 |          10 |             10 |          9 |
|       15 |       3 |      1109 |          10 |             10 |         10 |
|       16 |       4 |      3306 |          10 |             10 |         11 |
|       17 |       4 |      3306 |          10 |             10 |         12 |
|       18 |       4 |      3306 |          10 |             10 |         13 |
|       19 |       5 |      3304 |          10 |              2 |         11 |
|       20 |       5 |      3304 |          10 |              7 |         12 |
|       21 |       5 |      3304 |          10 |              3 |         13 |
|       22 |       6 |      2203 |          10 |             10 |          1 |
|       23 |       6 |      2203 |          10 |             10 |          2 |
|       24 |       6 |      2203 |          10 |             10 |          3 |
|       25 |       6 |      2203 |          10 |             10 |          4 |
|       26 |       6 |      2203 |          10 |             10 |          5 |
|       27 |       7 |      8909 |          40 |             35 |         14 |
+----------+---------+-----------+-------------+----------------+------------+
27 rows in set (0.00 sec)

mysql>
```

Notices Table:

```
mysql> select * from notices;
+-----------+-------------+-------------------------------------+
| notice_id | notice_date | notice_details                      |
+-----------+-------------+-------------------------------------+
|         1 | 2024-05-31  | Tomorrow all classes are cancelled! |
+-----------+-------------+-------------------------------------+
1 row in set (0.00 sec)

mysql>
```

Parents Table:

```
mysql> select * from parents;
+-----------+---------------+------------+---------------+--------+
| parent_id | parent_name   | student_id | email         | phone  |
+-----------+---------------+------------+---------------+--------+
|         1 | Nafees_parent |          4 | omi@gmail.com | 232323 |
|         2 | Tahsin_parent |          1 | omi@gmail.com | 232323 |
+-----------+---------------+------------+---------------+--------+
2 rows in set (0.00 sec)
```

Students Table:

```
mysql> select * from students;
+------------+-------------+-----------------+----------------------------------------+---------------+
| Student_ID | Student_Name | Enrollment_Status | department                             | department_id |
+------------+-------------+-----------------+----------------------------------------+---------------+
|          1 | Omi         | Enrolled        | Information Communication and Technology |             1 |
|          2 | Zawad       | Enrolled        | Information Communication and Technology |             1 |
|          3 | Asif        | Enrolled        | Information Communication and Technology |             1 |
|          4 | Nafees      | Enrolled        | Information Communication and Technology |             1 |
|          5 | Tahsin      | Enrolled        | Information Communication and Technology |             1 |
|          6 | Nibir       | Enrolled        | Computer Science and Engineering       |             2 |
|          7 | Rafsan      | Enrolled        | Computer Science and Engineering       |             2 |
|          8 | Ramisa      | Enrolled        | Computer Science and Engineering       |             2 |
|          9 | Lamia       | Enrolled        | Computer Science and Engineering       |             2 |
|         10 | Paromita    | Enrolled        | Computer Science and Engineering       |             2 |
|         11 | Joshua      | Enrolled        | Environmental Science                  |             3 |
|         12 | Shafin      | Enrolled        | Environmental Science                  |             3 |
|         13 | Maliha      | Enrolled        | Environmental Science                  |             3 |
|         14 | Niloy       | Enrolled        | Finance & Banking                      |          9909 |
|         15 | Imran       | Enrolled        | Computer Science and Engineering       |             2 |
+------------+-------------+-----------------+----------------------------------------+---------------+
15 rows in set (0.00 sec)

mysql>
```

Users Table:

```
mysql> select * from users;
+---------+---------------+----------+---------+--------------------------+
| user_id | username      | password | role    | picture                  |
+---------+---------------+----------+---------+--------------------------+
|       1 | admin         | 123      | admin   | images/admin.jpg         |
|       2 | Omi           | 123      | student | images/Omi.jpg           |
|       3 | ict_faculty1  | 123      | faculty | images/ict_faculty1.jpg  |
|       4 | ict_faculty2  | 123      | faculty | images/ict_faculty2.jpg  |
|       5 | ict_faculty3  | 123      | faculty | images/ict_faculty3.jpg  |
|       6 | ict_faculty4  | 123      | faculty | images/ict_faculty4.jpg  |
|       7 | ict_faculty5  | 123      | faculty | images/ict_faculty5.jpg  |
|       8 | Zawad         | 123      | student | images/Zawad.jpg         |
|       9 | Asif          | 123      | student | images/Asif.jpg          |
|      10 | Nafees        | 123      | student | images/Nafees.jpg        |
|      11 | Tahsin        | 123      | student | images/Tahsin.jpg        |
|      12 | cse_faculty1  | 123      | faculty | images/cse_faculty1.jpg  |
|      13 | cse_faculty2  | 123      | faculty | images/cse_faculty2.jpg  |
|      14 | cse_faculty3  | 123      | faculty | images/cse_faculty3.jpg  |
|      15 | Nibir         | 123      | student | images/Nibir.jpg         |
```

5.Pattern Identification Queries:

1. Attendance Patterns by Day of the Week:

SELECT DAYOFWEEK(attendance_date) AS Day, AVG(Status = 'Present') AS Average_Attendance

FROM attendance

GROUP BY DAYOFWEEK(attendance_date)

ORDER BY Average

2. Average Marks by Semester

SELECT e.Semester, AVG(m.Obtained_Marks) AS Average_Marks

FROM marks m

JOIN enrollment_status e ON m.Student_ID = e.Student_ID AND m.Course_ID = e.Course_ID

GROUP BY e.Semester

ORDER BY e.Semester;

3. Faculty Activity Patterns

SELECT c.course_faculty_id AS Faculty_ID, COUNT(*) AS Number_of_Submissions

FROM marks m

JOIN courses c ON m.Course_ID = c.Course_ID

GROUP BY c.course_faculty_id

ORDER BY Number_of_Submissions DESC;

4. Student Average Attendance Patterns

SELECT s.Student_Name, AVG(a.Status = 'Present') AS Average_Attendance

FROM attendance a

JOIN students s ON a.Student_ID = s.Student_ID

GROUP BY s.Student_Name

```
ORDER BY s.Student_Name;
```

5. Department Popularity

```
SELECT d.department_name, COUNT(s.Student_ID) AS Student_Count
FROM department d
JOIN students s ON d.department_id = s.department_id
GROUP BY d.department_name
ORDER BY Student_Count DESC;
```

6. Average Marks Distribution by Course

```
SELECT c.Course_Name, AVG(m.Obtained_Marks) AS Average_Marks
FROM marks m
JOIN courses c ON m.Course_ID = c.Course_ID
GROUP BY c.Course_Name
ORDER BY c.Course_Name;
```

7. Department Performance

```
SELECT d.department_name, AVG(m.Obtained_Marks) AS Average_Marks
FROM marks m
JOIN students s ON m.Student_ID = s.Student_ID
JOIN department d ON s.department_id = d.department_id
GROUP BY d.department_name
ORDER BY Average_Marks DESC;
```

8. Exam Performance by Type

```
SELECT et.Exam_Type_Name, AVG(m.Obtained_Marks) AS Average_Marks
FROM marks m
JOIN examinations e ON m.Exam_ID = e.Exam_ID
```

JOIN exam_types et ON e.Exam_Type_ID = et.Exam_Type_ID

GROUP BY et.Exam_Type_Name

ORDER BY Average_Marks DESC;

These queries were used to identify patterns and trends in the EduSync project and were displayed on the `admin_patterns.php` page.

6.Progress Reports:

Summary:

First, we have converted the ER Diagram to the relational schema. After that, we have mentioned the relation between the foreign key and primary key. Then we have checked normalization for each table. Our target was to convert all the tables into 3NF. So, as per the conditions of the third normal form, the tables were already in third normal form. Then we have created the database and generated the tables using SQL commands. Finally, we have inserted individual records into each of the tables.

Group Progress Report

Reporting time: 02.06.2024

Team Member	Total Hours (period)	Remarks
Prithviraj Chowdhury (2254901101)	28.04.2024 07:00PM-10:00PM (03 hours)	Converting ER diagram to relational schema, table design, normalization, data entry for a table
	05.05.2024 07:30PM-10:30PM (03 hours)	Worked on creating tables, inserted data into tables, worked on writing summary
	12.05.2024 08:00PM-10:30PM (02 hours 30 mins)	Writing the explanation of why the decomposed tables are in 1NF, 2NF and 3NF
	21.05.2024 8:00 PM-10:30 PM (03 hours 10 mins)	Inserted data into tables, worked on table constraints

	30.05.2024 7:00PM-9:00PM (2 hours)	Reviewed and corrected relational schema, ensured data consistency
	01.06.2024 8:00 AM- 5:00 PM (10 hours)	Finalized documentation and prepared the project
Gazi Mohammad Abrar Zawad (2254901049)	30.04.2024 07:00PM-10:00PM (03 hours)	Worked on relational schema, created database, created table, added foreign key, inserted data, updated table's constraints
	07.05.2024 08:00PM-10:30PM (02 hours 30 mins)	Modified the Entity Relationship Diagram, Added foreign keys, updated table constraints
	14.05.2024 08:30PM-11:00PM (02 hours 30 mins)	Data entry in the tables, Helped in Relational Schema, Document Formatting
	21.05.2024 08:30PM-11:00PM (02 hours 30 mins)	Worked on normalization, ensured all tables are in 3NF
	28.05.2024 08:30PM-11:00PM (02 hours 10 mins)	Reviewed ER diagram and relational schema, made necessary adjustments
	01.06.2024 08:30PM-11:00PM (02 hours 10 mins)	Finalized documentation and ensured all tables have individual records through the project

Jablay Noor Rahman (2254901093)	01.05.2024 07:00PM-10:00PM (03 hours)	Reviewing ER Diagram concepts, attribute analysis, initial report drafting
	08.05.2024 07:15PM-09:45PM (02 hours 30 mins)	Detailed functionality descriptions, finalizing attributes and keys
	15.05.2024 07:30PM-10:00PM (01 hours 30 mins)	Inserted data into tables, reviewed and corrected table designs
	22.05.2024 07:30PM-10:00PM (02 hours 30 mins)	Studied normalization checks, ensured all tables meet 3NF conditions
	29.05.2024 08:30PM-11:00PM (02 hours 10 mins)	Helped in preparing the final project report and presentation slides
	01.06.2023 02:00PM-5:00PM (03 hours)	Final report compilation, ER diagram review, preparing presentation materials
Group Totals	53 hours 10 mins	

Individual Report:

Member Name: Prithviraj Chowdhury Reporting Period: 02.06.2024

Date	Hours	Activities
28.04.2024	2 hours	Converting ER diagram to relational schema, table design
05.05.2024	2 hours 30 mins	Normalization, data entry for tables, writing explanation of why decomposed tables are in 1NF,2NF and 3NF
12.05.2024 15.05.2024 18.05.2024 23.05.2024 25.05.2024	18 hours 45 mins	Group meetings and discussion
19.05.2024	3 hours 10 mins	Inserted data into tables, worked on table constraints
26.05.2024	2 hours	Reviewed and corrected relational schema, ensured data consistency
01.06.2024	1 hour	Finalized documentation and prepared the project report

Member Name: Gazi Mohammad Abrar Zawad Reporting Period: 02.06.2024

Date	Hours	Activities
30.04.2024	2 hours	Created database and tables using SQL commands
07.05.2024	2 hours 30 mins	Inserted data into tables, helped in writing the summary
12.05.2024 15.05.2024 18.05.2024 23.05.2024 25.05.2024	18 hours 45 mins	Group meetings and discussion
21.05.2024	3 hours 10 mins	Worked on normalization, ensured all tables are in 3NF
28.05.2024	2 hours 40 mins	Reviewed ER diagram and relational schema, made necessary adjustments
01.06.2024	1 hour	Finalized documentation and ensured all tables have individual records

Member Name: Jablay Noor Rahman Reporting Period: 02.06.2024

Date	Hours	Activities

30.04.2024	3 hours	Studied constraints and applied them to tables
01.05.2024	1 hours 30 mins	Worked on converting ER diagram to relational schema, ensured all relationships are correctly adjusted
12.05.2024 15.05.2024 18.05.2024 23.05.2024 25.05.2024	18 hours 45 mins	Group meetings and discussion
21.05.2024	1 hours 10 mins	Inserted data into tables, reviewed and corrected table designs
28.05.2024	2 hours 20 mins	Wrote explanations for normalization checks, ensured all tables meet 3NF conditions
29.05.2024	2 hours 10 mins	Prepared Presentation Slides and helped in documentation
01.06.2024	1 hour	Final review and submission preparation

Learning Outcomes of the Project:

1. Technologies and Skills:

PHP:

Server-Side Scripting: We gained proficiency in using PHP for server-side scripting, enabling dynamic content generation and interaction with the database.

Form Handling: We learned how to handle form submissions, validate user inputs, and provide meaningful feedback to users.

Sessions and Cookies: We understood the importance of maintaining state using sessions and cookies, essential for user authentication and personalization.

Error Handling: Implementing try-catch blocks and custom error messages improved our ability to manage errors gracefully and debug effectively.

MySQL:

Database Design: We learned to design a relational database from scratch, starting from the ER diagram to the relational schema.

Normalization: Ensuring our tables adhered to 3NF helped us understand normalization principles and their importance in reducing redundancy and improving data integrity.

SQL Queries: We became adept at writing complex SQL queries for data retrieval, insertion, updating, and deletion. Joins, subqueries, and aggregate functions were particularly useful.

Foreign Keys and Constraints: We learned how to define foreign keys and enforce referential integrity, ensuring that relationships between tables were maintained correctly.

HTML & CSS:

Structuring Web Pages: We learned to structure web pages using HTML, employing semantic elements to improve accessibility and to make our pages visually pleasing.

Styling: CSS enabled us to create visually appealing and responsive designs. We explored various selectors, properties, and layout techniques such as Flexbox and Grid.

Form Elements: Creating and styling form elements like text inputs, dropdowns, and buttons enhanced our ability to collect and present data effectively.

2. Features and Explanations

User Authentication:

Login/Logout System: We created a secure login/logout system using PHP sessions to manage user authentication. Passwords were hashed for security.

Role-Based Access: Different user roles (admin, student, faculty, parent) were assigned, each having access to specific functionalities and dashboards.

Dashboards:

Admin Dashboard: Features for managing courses, students, faculties, and notices. Admins can add, update, delete records, and generate reports.

Faculty Dashboard: Faculty members can manage attendance, provide marks, and can add new courses according to their preference.

Student Dashboard: Students can view their courses, marks, attendance, and notices.

Parent Dashboard: Parents can view their children's attendance and marks, ensuring they stay informed about their academic progress.

Attendance and Marks Management:

Attendance Tracking: Faculty can record attendance, and both students and parents can view attendance records.

Marks Management: Faculty can enter marks for different exams, which students and parents can view.

Notices and Announcements:

Notice Board: Admins can post notices, which are visible to all users based on their roles. Students and parents can stay updated with important announcements.

3.Practical Applications of This Project

Educational Institutions:

Streamlined Administration: The system can be used by educational institutions to streamline administrative tasks, reducing paperwork and manual errors.

Parent Engagement: By providing parents with access to their children's academic data, schools can foster better parent-teacher collaboration and engagement.

Student Monitoring: Students can monitor their academic performance, attendance, and receive timely updates, promoting self-regulation and responsibility.

Corporate Training Programs:

Employee Training: Companies can use similar systems to manage employee training programs, track attendance, and performance in various courses and certifications.

Online Learning Platforms:

Course Management: Online learning platforms can utilize this system to manage courses, track student progress, and facilitate communication between instructors and students.

4. Limitations of This Project:

Scalability:

Database Performance: As the number of users and data volume increases, the performance of the database may degrade. Indexing and database optimization techniques would be required.

Load Handling: The current implementation may not handle a large number of simultaneous users effectively. Implementing load balancing and server optimization would be necessary.

Security:

Basic Security Measures: While basic security measures like password hashing and session management are implemented, advanced security practices like SQL injection prevention, cross-site scripting (XSS) protection, and secure password storage (e.g., using bcrypt) should be employed.

Data Privacy: Ensuring compliance with data privacy regulations (e.g., GDPR) is crucial, and our current system might need enhancements to meet these standards.

User Experience:

Basic UI/UX: The user interface, while functional, may need significant improvements to enhance the user experience. A more intuitive and visually appealing design would be beneficial.

Mobile Responsiveness: Although CSS was used to create responsive designs, further testing and optimization for various devices and screen sizes are required.

Features:

Limited Features Some advanced features like real-time notifications, automated reminders, and advanced analytics are not included but would greatly enhance the system's functionality.

Customizability: The system is currently designed for a specific educational context. Making it more customizable to fit different institutional needs and configurations would increase its applicability.

Conclusion

The development of the EduSync project has been a comprehensive learning experience, covering various aspects of web development and database management. We have built a functional and efficient system that addresses the core needs of educational institutions, while also identifying areas for future improvement and expansion. The knowledge and skills gained from this project are invaluable and provide a solid foundation for tackling more complex projects in the future.