



# Database Management Project

## Project Report

### GROUP-13

NAME	ID
Md. Ezaz Ahamed	2021757
Roshni Parvin	2030063
Md Tasin Rahman	2031091
Samiul Islam Sourav	2031109
Hasan Imam	2031121
Toufikul Islam Eyasin	2031254

## Table of Contents

CHAPTER-1 INTRODUCTION .....	3
BACKGROUND OF THE ORGANIZATION – IUB: .....	3
I. BACKGROUND OF THE PROJECT .....	4
II. OBJECTIVE OF THE PROJECT.....	5
III. SCOPE OF THE PROJECT.....	5
CHAPTER-2 REQUIREMENT ANALYSIS.....	6
EXISTING SYSTEM:.....	6
i. RICH PICTURE- EXISTING SYSTEM.....	6
ii. SIX ELEMENT ANALYSIS – EXISTING SYSTEM .....	8
I. PROBLEM ANALYSIS – EXISTING SYSTEM.....	29
Proposed System:.....	32
I. RICH PICTURE – PROPOSED SYSTEM.....	32
II. SIX ELEMENT ANALYSIS – PROPOSED SYSTEM.....	33
CHAPTER – 3 LOGICAL SYSTEM DIAGRAM.....	56
I. BUSINESS RULES – PROPOSED SYSTEM .....	56
II. ENTITY RELATION DIAGRAM:.....	59
III. ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL SCHEMA: .....	60
IV. NORMALIZATION .....	61
V. DATA DICTIONARY .....	73
CHAPTER-4 PHYSICAL SYSTEM DIAGRAM.....	95
I. INPUT FORMS .....	95
II. OUTPUT FORMS .....	105
CHAPTER – 5 CONCLUSION.....	118
PROBLEM AND SOLUTION.....	118
I. Analysis Phase:.....	118
II. Designing Phase:.....	118
III. Implementation Phase: .....	118
IV. Additional features and Future Development:.....	119

## CHAPTER-1 INTRODUCTION

### BACKGROUND OF THE ORGANIZATION – IUB:

In 1993, the private IUB university was founded in Bangladesh. There are now over 10,000 undergraduate and graduate students enrolled there. More than 450 incredibly talented and highly qualified faculty members, at least half of them possess PhDs, who excel in their fields of instruction. The number of IUB alumni is close to 14,000. IUB currently has 5 academic institutions.

- Business & Entrepreneurship
- Engineering, Technology & Sciences
- Environment and Life Sciences
- Liberal Arts & Social Sciences
- Pharmacy and Public Health

The institution actively aided in the growth of Bangladesh's education sector and produced capable, knowledgeable scholars who have made contributions on both a national and international scale. Regular curriculum updates, the implementation of a system to track student performance based on a quantified approach between course curriculum and standards set by UGC and the Bangladesh government, the Ministry of Education, and other necessary institutions for each of the schools, as well as ongoing student performance monitoring, have all assisted IUB in achieving this.



Figure 1: IUB at a glance

The goals of IUB are to produce local graduates who meet international standards and have the knowledge and skills to lead in business, government, and welfare. They also include encouraging and supporting useful research, creating knowledge, and providing opportunities for adults to continue their education.

## I. BACKGROUND OF THE PROJECT

A framework for outcome-based education (OBE) is the Student Performance Monitoring System. It assesses students, faculty, school, department, and program performance and aids the higher authorities of the educational institution in developing strategies for enhancements, including the deployment of a skill and spider chart. The purpose of our project is to design, develop, and disseminate software that, in our opinion, will help colleges all over the world encourage a more effective and beneficial technique of student evaluation. The idea of course outcomes (COs) and program learning outcomes (PLOs), where each CO is mapped to a PLO, and each PLO represents a specific valuable skill that students are expected to acquire or improve at the conclusion of that course, such as problem analysis, design, implementation of a skill, and spider chart, has been introduced

as the central concept of our project. The project will also help to input the information manually using a google form.

The CO will be determined using the program which is eventually connected with the PLOs. Later, the grades will be analyzed according to the CO rate. The program will help the authority and faculties to evaluate the student's achievement out of the outcomes the course has provided them with. Students can keep track of their development in each area and pinpoint their areas for development. Our application also intends to assist institutional organizations, such as academic, administrative, and departmental bodies, in better distributing and allocating resources while keeping track of student development and departmental performance.

## II. OBJECTIVE OF THE PROJECT

The main intention of the program is to analyze if a student has achieved a certain CO (Course Outline) and PLO (Program Learning Outcome) or not. In this project details of the students can be input manually also through a google form which will be connected. Here, a database is created and using the marks of the students it is ensured whether they have learned the PLOs of a certain program. Depending on the PLOs, the percentage of CO is also determined using the data. The uniform set of data will help the university authorities, faculties and other related staff to judge and evaluate the progress of the student in a certain program.

## III. SCOPE OF THE PROJECT

We thoroughly analyzed the current system and found a few problems that could slow down corporate operations, make them less effective, and lead to communication breakdowns.

A more advanced system that uses a Relational Database Management System (RDBMS) to store, update, and retrieve necessary documents like course outlines, exam question papers, and answer scripts as well as other necessary data required to monitor student performance and produce other OBE (outcome-Based Education) reports is the proposed solution to resolve those issues.

We've identified every user of the system, how they'll obtain the information they need, how they'll communicate with one another, etc.

We want to create user interfaces so that every user can access the data they need and utilize the system to create, read, and download the reports and documents they want.

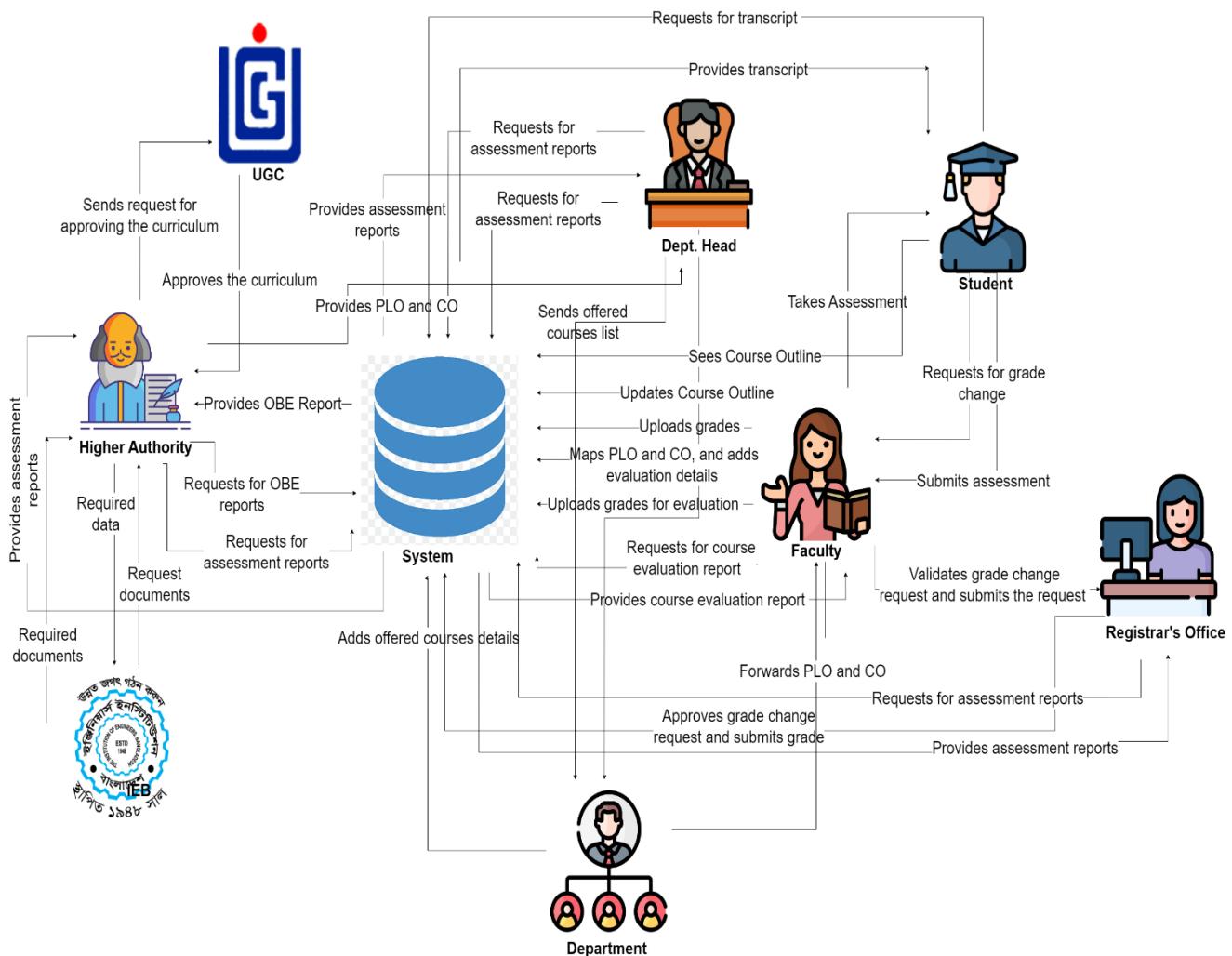
## CHAPTER-2 REQUIREMENT ANALYSIS

The process of figuring out what the database is utilized for is called requirement analysis. Interviews with stakeholders are conducted to ascertain the operations that must be carried out, the data that must be processed, and the functionality and system requirements they expect and require from the database. By doing this, we can fully comprehend the stakeholders and their interactions with one another.

### EXISTING SYSTEM:

#### i. RICH PICTURE- EXISTING SYSTEM

A technique to illustrate system processes that is simpler for everyone to understand is using rich graphics. It is made up of words, symbols, and icons that are all utilized to graphically depict the scenario. In order to see relationships, we need a rich image and linkages that we might overlook otherwise. It aids in determining one or more themes that participants might choose to investigate and address further. Vivid images are thus always employed during the pre-analysis stage.



In this rich picture, the stake holders are:

1. UGC
2. IEB
3. Higher Authority
4. Department head
5. Department office

6. Admin
7. Registrar's office
8. Faculty
9. Student

**The main storage is:**

1. SPMS

## ii. SIX ELEMENT ANALYSIS – EXISTING SYSTEM

The main processes are seen from the detailed picture as follows:

- 1) Creating storing and giving Course Outline
- 2) Add Questions to the question bank and grading the answer script
- 3) Course based student performance trend according to GPA
- 4) Faculty based student performance according to GPA
- 5) Course wise PLO achievement of a student
- 6) Student performance trend under VC/Dean/Head of Department

- 7) Course, Program, department, school CLO-PLO statistics
- 8) Course, student, department school wise expected vs achieved PLO
- 9) Department average of total PLO achieved and attempted students
- 10) Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.

The following six elements can be used in six element analysis to determine how they will affect a process:

1. Human
2. Non computing Hardware
3. Computing Hardware
4. Software.
5. Database.
6. Network and Communication

Process	Human	Non-computing Hardware	Computing Hardware	Software	Database	Network
<b>Creating, storing, and giving Course outline</b>	<b>Faculty:</b> a. At first the faculty will login into the system using ID		<b>Computer/ Laptop</b> a. It is used to login into the system.	<b>SPMS 3.0</b> a. Employed to store data in the database.	<b>SPMS 3.0 Database</b> a. All valid data are stored here which can be modified	<b>Internet</b> a. Used to sign in into the system

	<p>and password.</p> <p>b. Select the option to create course outline.</p> <p>c. Add course outline and upload it.</p> <p>d. Store the outline in the system.</p> <p><b>Students:</b></p> <p>a. Students will login into their account using their ID and password.</p> <p>b. Then select the course outline option.</p>	<p><b>Printer:</b></p> <p>a. This is used to get the hard copy of the outline when needed.</p>		<p>by the admin when needed.</p> <p>and browse</p>
--	--	--	--	--

	c. They can view/ download the option.					
<b>Add Questions to the question bank and grading the answer script</b>	<p><b>Faculty</b></p> <p>a. Enters their ID and password to log in.</p> <p>b. Choose the course and the sections where you must provide an answer.</p> <p>c. Enter the query in the query bank.</p> <p>d. Choose the Assign option.</p> <p>e. Evaluate the responses that the students submitted.</p> <p><b>Students:</b></p>	<p><b>Computer/ Laptop:</b></p> <p>a. Used as a Sign-In for SPMS 2.0</p> <p><b>Printer:</b></p> <p>a. Applied to printing the section's overall grades.</p>	<p><b>SPMS 3.0:</b></p> <p>a. Used to generate a result graph utilizing data from the database or to save data in the database.</p>	<p><b>SPMS 3.0 Database</b></p> <p>a. All legitimate data are kept here and can be updated by the administrators.</p>	<b>Internet</b>	Used to sign to SPMS 3.0.

	<p>a. Enters their ID and password to log in.</p> <p>b. Respond to the faculty-assigned question in the answer database.</p> <p>c. Choose the "Submit" option.</p> <p>d. Verify your grade in SPMS 3.0 when the instructor has finished.</p>				
<b>Course based student performance trend according to GPA</b>	<p><b>Department Head:</b></p> <p>a. Enters their ID and password to log in.</p> <p>b. Enter the viewing time</p>	<p><b>Computer/ Laptop</b></p> <p>a. Used to Sign into SPMS 3.0.</p>	<p><b>SPMS 3.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database</p>	<p><b>SPMS 3.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrat</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 3.0.</p>

	<p>frame and course ID.</p> <p>c. Examine student progress using a graph created after analysis and the GPAs attained by the highest/lowest /average students.</p> <p><b>Faculty:</b></p> <p>A user a. enters their ID and password to access the system.</p> <p>b. Use the course ID and time period to find the course they are teaching, and then look at the students'</p>		<p><b>Printer</b></p> <p>b. Used to print hard copy of the progress of current semester's students and compare with the progress of the previous semester's students who did that course</p>	<p>or to save student data in the database.</p> <p>ors from the database.</p>	
--	--	--	--	---	--

	<p>progress in that course.</p> <p><b>Students:</b></p> <p>a. The user enters their ID and password to access the system.</p> <p>b. Use the course ID to search for the course, and then view the student's grade point average (GPA) and progress in that course.</p> <p><b>Dean/VC:</b></p> <p>a. Entering their ID and password to log in to the system.</p>				
--	---	--	--	--	--

	b. Use the course ID and time period to search for the course, then view the students' progress in that course.					
<b>Faculty based student performance according to GPA</b>	<p><b>Faculty:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Track the Development of the Learners under their Instruction.</p> <p><b>Head of Department:</b></p> <p>a. Entering their ID and password to</p>	<p><b>Computer/ Laptop:</b></p> <p>a. Used as a Sign-In for SPMS 3.0</p> <p><b>Printer:</b></p> <p>a. Applied to printing the hardcopy.</p>	<p><b>SPMS 3.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database</p>	<p><b>SPMS 3.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrators from the database.</p>	<b>Internet</b>	Used to sign to SPMS 3.0.

	<p>log in to the system.</p> <p>b. Use the faculty's name to find the professor that has to be evaluated.</p> <p>c. Evaluate the progress of the students being taught by the faculty based on the GPA they have attained.</p> <p><b>Dean/VC:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Use the faculty member's name and Department ID to find the</p>				
--	---	--	--	--	--

	<p>faculty member who will be evaluated.</p> <p>c. Based on the GPAs that the students have received, view the progress of the pupils who are being taught by that faculty.</p>				
<b>Course wise PLO achievement of a student</b>	<p><b>VC/Dean:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the PLO accomplishment tab and enter the course ID.</p> <p>c. Review the student's PLO</p>	<p><b>Computer/</b></p> <p><b>Laptop</b></p> <p>a. Used to Sign into SPMS 3.0</p> <p><b>Printer</b></p> <p>a. Used to print hard copy of a report of students who completed most the PLO</p>	<p><b>SPMS 3.0</b></p> <p><b>SPMS 3.0 Database</b></p> <p>a. It is used to store data and automatically generate PLOs based on the CO given.</p>	<p><b>SPMS 3.0 Database</b></p> <p>a. All current data are saved here and can be updated by administrators.</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 3.0.</p>

	accomplishments.	achievements if needed.			
	<p><b>Department Head:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the PLO accomplishment tab and enter the course ID.</p> <p>c. Examine the PLOs that the students met.</p> <p><b>Faculty:</b></p> <p>a. Enters their ID and password to log in to the system.</p>				

	<p>b. Click the PLO accomplishment tab and enter the course ID.</p> <p>c. Check out the PLOs that the students in a course have met.</p> <p><b>Student:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Review the PLOs they have already met and the number they still need to reach to pass the course.</p>				
<b>Student performance trend under</b>	<b>Dean:</b>		<b>Computer/ Laptop:</b>	<b>SPMS 3.0:</b>	<b>SPMS 3.0 database:</b>

<b>VC/Dean/Head of Department</b>	<p>a. Enters their ID and password to log in to the system.</p> <p>b. Using their Name and Department ID, look up the Department Head to be checked.</p> <p>c. Check out the students' development beneath them or them.</p> <p><b>VC:</b></p> <p>a. User logs in to system with ID and password.</p> <p>b. Use their name and either their school ID or department ID</p>		<p>a. Used as a Sign-In for SPMS 3.0</p> <p><b>Printer:</b></p> <p>a. Applied to printing the hardcopy.</p>	<p>a. Used to create graphs of statistical analysis using data from the database or to save student data in the database</p>	<p>a. All current information is kept here and can be updated by administrators from the database.</p>	Used to sign to SPMS 3.0.
-----------------------------------	--	--	---	--	--	---------------------------

	<p>to search for a dean or department head to be checked.</p> <p>c. Track student development underneath them.</p> <p><b>Department Head:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>c. Track students' development beneath them.</p>				
<b>Course, Program, department, school CLO-PLO statistics</b>	<p><b>Dean/VC:</b></p> <p>a. Enters their ID and password to log in to the system.</p>	<p><b>Computer/ Laptop:</b></p> <p>a. Used as a Sign-In for SPMS 3.0</p>	<p><b>SPMS 3.0:</b></p> <p>a. Used to create graphs of statistical analysis using data</p>	<p><b>SPMS 3.0 database:</b></p> <p>a. All current information is kept here and can be updated by</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 3.0.</p>

	<p>b. Check out the CLO-PLO mapping student achievement figures.</p> <p><b>Department Head:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Check out the CLO-PLO mapping student achievement figures.</p> <p><b>Faculty:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Check out the CLO-PLO mapping student</p>	<p><b>Printer:</b></p> <p>a. Applied to printing the hardcopy.</p>	<p>from the database or to save student data in the database</p>	<p>administrators from the database.</p>	
--	---	--	--	--	--

	<p>achievement figures.</p> <p><b>Student:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Examine the CLO-PLO mapping statistics that they and other students achieved.</p>				
<b>Course, student, department school wise expected vs achieved PLO</b>	<p><b>Dean/VC:</b></p> <p>a. Log in to the system with your ID and password.</p> <p>b. Compare the expected and realized PLOs for the students for the time that has been entered.</p>	<p><b>Computer/ Laptops:</b></p> <p>a. Used to sign into SPMS 3.0 Printer</p> <p>b. Used to print the attained PLO for the previous and current semesters on paper</p>	<p><b>SPMS 3.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database or to save student data in the database</p>	<p><b>SPMS 3.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrators from the database.</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 3.0.</p>

	<p><b>Department Head:</b></p> <p>a. Log in to the system with your ID and password.</p> <p>b. Compare the expected and realized PLOs for the students for the time that has been entered.</p> <p><b>Faculty:</b></p> <p>a. Log in to the system using your ID and password.</p> <p>b. Compare the expected and realized PLOs for the students for the time that has been entered.</p>	for comparison .			
--	--	------------------	--	--	--

	<b>Student:</b>  a. Log in to the system using your ID and password.  b. Compare the expected and realized PLOs for the students for the time that has been entered.				
<b>Department average of total PLO achieved and attempted students</b>	<b>Dean/VC:</b>  a. Enter your ID to access the system and use your password.  b. Choose the semester's desired viewing time range.  c. See the overall PLO average for the department as	<b>Computer/ Laptops:</b>  a. Used to sign into SPMS 3.0  b. Used to print the attained PLO for the previous and current semesters on paper for	<b>SPMS 3.0:</b>  a. Used to create graphs of statistical analysis using data from the database or to save student data in the database	<b>SPMS 3.0 database:</b>  a. All current information is kept here and can be updated by administrators from the database.	<b>Internet</b>  Used to sign to SPMS 3.0.

	<p>well as the number of students that attempted.</p> <p><b>Department Head:</b></p> <p>a. Log in to the system using your ID and password.</p> <p>b. Choose the semester's desired viewing time range.</p> <p>c. See the overall PLO average for the department as well as the number of students that attempted.</p> <p><b>Faculty:</b></p> <p>a. Enter your ID and</p>	comparison .			
--	---	--------------	--	--	--

	<p>password to log in.</p> <p>b. Check out the students' PLO average across the entire department.</p> <p><b>Student:</b></p> <p>a. Log in to the system with your ID and password.</p> <p>b. See the students' overall departmental average of PLO results.</p>				
<b>Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.</b>	<p><b>VC:</b></p> <p>1) Sign into the system using ID and Password.</p> <p>2) Select Student Enrollment Statistics tab</p>	<p><b>Computer/ Laptops:</b></p> <p>a. Used to sign into SPMS 3.0 Printer</p> <p>b. Used to print the attained</p>	<p><b>SPMS 3.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database</p>	<p><b>SPMS 3.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrat</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 3.0.</p>

	<p>and select Year and Semester under that tab</p> <p>3) View Student Enrollment Statistics Of That Year and Semester.</p> <p><b>Dean:</b></p> <p>1) Sign into the system using ID and Password.</p> <p>2) Select Student Enrollment Statistics tab and select Year and Semester under that tab</p> <p>3) View Student Enrollment Statistics Of that Year and Semester.</p>		<p>PLO for the previous and current semesters on paper for comparison .</p>	<p>or to save student data in the database</p>	<p>ors from the database.</p>	
--	---	--	---	--	-------------------------------	--

	<p><b>Department Head:</b></p> <p>1) Sign into the system using ID and Password.</p> <p>2) Select Student Enrollment Statistics tab and select Year and Semester under that tab</p> <p>3) View Student Enrollment Statistics Of that Year and Semester</p>				
--	--	--	--	--	--

## I. PROBLEM ANALYSIS – EXISTING SYSTEM

The problems in the existing system were analyzed and the probable solutions are mentioned.

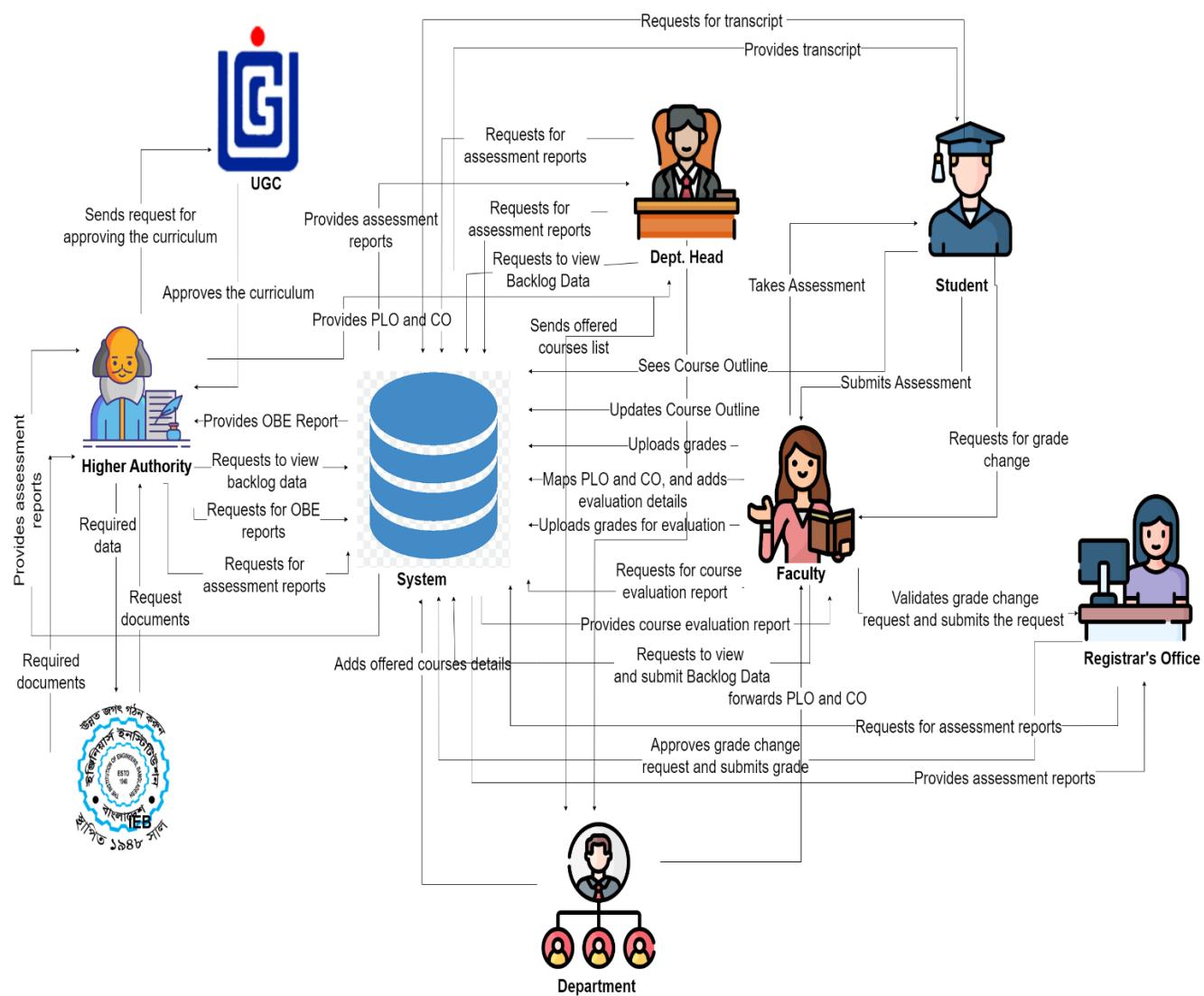
Process Name	Stake Holder	Concerns (Problems)	Analysis (Reason of the problem)	Proposed Solution

<b>Student Evaluation</b>	Faculty:  Student:	1.The Course Outcome (CO) is missing in the evaluation.  2.The achievements of students in courses cannot be measured accurately.	The Program Learning Outcome (PLO) and Course Learning Outcome (CO) are related which is why often they are confused with each other. But the CO are a part of the PLO which needs to be identified separately to properly evaluate the students.	The CO should also be sent to the faculties from higher authorities to the faculties. Then the CO of the course should be mapped. Then the PLO should be analyzed using the COs. Then the final grade should be analyzed.
<b>Grade Submission</b>	Faculty:	1.Here, the faculty can only input the grades and details of the student manually.	The reason of the problem can be that this method has been preferred traditionally by the faculties and authorities.	The new project can be developed which will use a form to take the details of the student and grades which will directly be exported by the project in database.

<b>Backlog Data</b>	<p>Faculty:</p> <p>Higher Authority:</p>	<p>1. In the current system, there is no existence of the backlog data.</p> <p>2. The lack of database makes it tough to manage the data of the students.</p>	<p>Mostly, in the earlier system the details were stored in physical storage. Along with that, the outcomes were not evaluated. Also, the faculties manually added the information of the students.</p>	<p>In the new project, the faculty can manually add the data which will be stored in the database for later use. Also the csv file can be used to import the information. The backlog data will be useful to store the information for later use. This will also help to create the evaluation.</p>
---------------------	--	---	---	---

## PROPOSED SYSTEM:

### I. RICH PICTURE – PROPOSED SYSTEM



In this rich picture, the stake holders are:

1. UGC
2. IEB
3. Higher Authority
4. Department head

5. Department office
6. Admin
7. Registrar's office
8. Faculty
9. Student

The main storage is:

1. SPMS

## II. SIX ELEMENT ANALYSIS – PROPOSED SYSTEM

**The main processes are seen from the detailed picture as follows:**

- 1) Creating storing and giving Course Outline
- 2) Add Questions to the question bank and grading the answer script
- 3) Course based student performance trend according to GPA
- 4) Faculty based student performance according to GPA
- 5) Course wise PLO achievement of a student
- 6) Course wise CO achievement of a student
- 7) Student performance trend under VC/Dean/Head of Department
- 8) Course, Program, department, school CO-PLO statistics
- 9) Course, student, department school wise expected vs achieved PLO and CO.

10) Department average of total PLO and CO achieved and attempted students

11) Student Enrollment Statistics VC-wise, Dean-wise, Department

Head-wise.

**The following six elements can be used in six element analysis to determine how they will affect a process:**

1. Human
2. Non computing Hardware
3. Computing Hardware
4. Software.
5. Database.
6. Network and Communication

Process	Human	Non-computing Hardware	Computing Hardware	Software	Database	Network
<b>Creating, storing, and giving Course outline</b>	<b>Faculty:</b> e. At first the faculty will login into the system using ID		<b>Computer/Laptop</b> a. It is used to login	<b>SPMS 4.0</b> a. Employed to store	<b>SPMS 4.0 database</b> a. All valid data are stored here	<b>Internet</b> a. Used to sign in

	<p>and password.</p> <p>f. Select the option to create course outline.</p> <p>g. Add course outline and upload it.</p> <p>h. Store the outline in the system.</p> <p><b>Students:</b></p> <p>a. Students will login into their account using their ID and password.</p> <p>b. Then select the course outline option.</p> <p>c. They can view/ download the option.</p>	<p>into the system.</p> <p><b>Printer:</b></p> <p>a. This is used to get the hard copy of the outline when needed.</p>	<p>data in the database.</p> <p>which can be modified by the admin when needed.</p>	<p>into the system and browse</p>
--	--	--	---	-----------------------------------

Add Questions to the question bank and grading the answer script	Faculty	Computer/Laptop:	SPMS 4.0:	SPMS 4.0 Database	Internet
	<p>a. Enters their ID and password to log in.</p> <p>b. Choose the course and the sections where you must provide an answer.</p> <p>c. Enter the query in the query bank.</p> <p>d. Choose the Assign option.</p> <p>e. Evaluate the responses that the students submitted.</p> <p><b>Students:</b></p> <p>a. Enters their ID and password to log in.</p>	<p><b>Printer:</b></p> <p>a. Used as a Sign-In for SPMS 2.0</p> <p>a. Applied to printing the section's overall grades.</p>	<p>a. Used to generate a result graph utilizing data from the database or to save data in the database.</p>	<p>a. All legitimate data are kept here and can be updated by the administrators.</p>	Used to sign to SPMS 4.0.

	<p>b. Respond to the faculty-assigned question in the answer database.</p> <p>c. Choose the "Submit" option.</p> <p>d. Verify your grade in SPMS 3.0 when the instructor has finished.</p>					
<b>Course based student performance trend according to GPA</b>	<p><b>Department Head:</b></p> <p>a. Enters their ID and password to log in.</p> <p>b. Enter the viewing time frame and course ID.</p> <p>c. Examine student progress using a graph created after analysis</p>	<p><b>Computer/ Laptop</b></p> <p>a. Used to Sign into SPMS 4.0.</p> <p><b>Printer</b></p> <p>b. Used to print hard copy of the</p>	<p><b>SPMS 4.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database or to save student data in the database.</p>	<p><b>SPMS 4.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrators from the database.</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 4.0.</p>	

	<p>and the GPAs attained by the highest/lowest/ average students.</p> <p><b>Faculty:</b></p> <p>A user a. enters their ID and password to access the system.</p> <p>b. Use the course ID and time period to find the course they are teaching, and then look at the students' progress in that course.</p> <p><b>Students:</b></p> <p>a. The user enters their ID and password</p>	<p>progress of current semester's students and compare with the progress of the previous semester's students who did that course</p>			
--	--	--	--	--	--

	<p>to access the system.</p> <p>b. Use the course ID to search for the course, and then view the student's grade point average (GPA) and progress in that course.</p> <p><b>Dean/VC:</b></p> <p>a. Entering their ID and password to log in to the system.</p> <p>b. Use the course ID and time period to search for the course, then view the students' progress in that course.</p>				
--	---	--	--	--	--

<b>Faculty based student performance according to GPA</b>	<b>Faculty:</b>  a. Enters their ID and password to log in to the system.  b. Track the Development of the Learners under their Instruction.	<b>Computer/ Laptop:</b>  a. Used as a Sign-In for SPMS 4.0	<b>SPMS 4.0:</b>  a. Used to create graphs of statistical analysis using data from the database	<b>SPMS 4.0 database:</b>  a. All current information is kept here and can be updated by administrators from the database.	<b>Internet</b>
	<b>Head of Department:</b>  a. Entering their ID and password to log in to the system.  b. Use the faculty's name to find the professor that has to be evaluated.  c. Evaluate the progress of the students being	<b>Printer:</b>  a. Applied to printing the hardcopy.			

	<p>taught by the faculty based on the GPA they have attained.</p> <p><b>Dean/VC:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Use the faculty member's name and Department ID to find the faculty member who will be evaluated.</p> <p>c. Based on the GPAs that the students have received, view the progress of the pupils who are being</p>				
--	---	--	--	--	--

	taught by that faculty.					
<b>Course wise PLO achievement of a student</b>	<p><b>VC/Dean:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the PLO accomplishment tab and enter the course ID.</p> <p>c. Review the student's PLO accomplishments.</p> <p><b>Department Head:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the PLO</p>	<p><b>Computer/ Laptop</b></p> <p>a. Used to Sign into SPMS 4.0</p> <p><b>Printer</b></p> <p>a. Used to print hard copy of a report of students who completed most the PLO achievements if needed.</p>	<p><b>SPMS 4.0</b></p> <p>a. It is used to store data and automatically generate PLOs based on the CO given.</p>	<p><b>SPMS 4.0 Database</b></p> <p>a. All current data are saved here and can be updated by administrators.</p>	<p><b>SPMS 4.0</b></p> <p>a. All current data are saved here and can be updated by administrators.</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 4.0.</p>

	<p>accomplishment tab and enter the course ID.</p> <p>c. Examine the PLOs that the students met.</p> <p><b>Faculty:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the PLO accomplishment tab and enter the course ID.</p> <p>c. Check out the PLOs that the students in a course have met.</p> <p><b>Student:</b></p> <p>a. Enters their ID and</p>				
--	---	--	--	--	--

	<p>password to log in to the system.</p> <p>b. Review the PLOs they have already met and the number they still need to reach to pass the course.</p>					
<b>Course wise CO achievement of a student</b>	<p><b>VC/Dean:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the CO accomplishment tab and enter the course ID.</p> <p>c. Review the student's CO accomplishments.</p> <p><b>Department Head:</b></p>	<p><b>Computer/ Laptop</b></p> <p>a. Used to Sign into SPMS 4.0</p> <p><b>Printer</b></p> <p>a. Used to print hard copy of a report of students who completed most the CO achievements.</p>	<p><b>SPMS 4.0</b></p> <p>a. It is used to store data and automatically generate COs based on the course outline given.</p>	<p><b>SPMS 4.0 Database</b></p> <p>a. All current data are saved here and can be updated by administrators.</p>	<p><b>Internet</b></p>	

	<p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the CO accomplishment tab and enter the course ID.</p> <p>c. Examine the COs that the students met.</p> <p><b>Faculty:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Click the CO accomplishment tab and enter the course ID.</p> <p>c. Check out the COs that the students in</p>	nts if needed.			
--	--	----------------	--	--	--

	<p>a course have met.</p> <p><b>Student:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Review the COs they have already met and the number they still need to reach to pass the course.</p>				
<b>Student performance trend under VC/Dean/Head of Department</b>	<p><b>Dean:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Using their Name and Department ID, look up the Department Head to be checked.</p>	<p><b>Computer/Laptop:</b></p> <p>a. Used as a Sign-In for SPMS 4.0</p> <p><b>Printer:</b></p> <p>a. Applied to printing the hardcopy.</p>	<p><b>SPMS 4.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database or to save student data in the database.</p>	<p><b>SPMS 4.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrators from the database.</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 4.0.</p>

	<p>c. Check out the students' development beneath them or them.</p> <p><b>VC:</b></p> <p>a. User logs in to system with ID and password.</p> <p>b. Use their name and either their school ID or department ID to search for a dean or department head to be checked.</p> <p>c. Track student development underneath them.</p> <p><b>Department Head:</b></p>				
--	--	--	--	--	--

	<p>a. Enters their ID and password to log in to the system.</p> <p>c. Track students' development beneath them.</p>				
<b>Course, Program, department, school CLO-PLO statistics</b>	<p><b>Dean/VC:</b></p> <p>a. Enters their ID and password to log in to the system.</p> <p>b. Check out the CO-PLO mapping student achievement figures.</p> <p><b>Department Head:</b></p> <p>a. Enters their ID and password to log in to the system.</p>	<p><b>Computer/Laptop:</b></p> <p>a. Used as a Sign-In for SPMS 4.0</p> <p><b>Printer:</b></p> <p>a. Applied to printing the hardcopy.</p>	<p><b>SPMS 4.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database</p> <p>a. Applied to printing the hardcopy.</p>	<p><b>SPMS 4.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrators from the database.</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 4.0.</p>

	b. Check out the CO-PLO mapping student achievement figures.  <b>Faculty:</b>  a. Enters their ID and password to log in to the system.  b. Check out the CO-PLO mapping student achievement figures.  <b>Student:</b>  a. Enters their ID and password to log in to the system.					
--	--	--	--	--	--	--

	b. Examine the CO-PLO mapping statistics that they and other students achieved.					
<b>Course, student, department school wise expected vs achieved PLO and CO</b>	<p><b>Dean/VC:</b></p> <p>a. Log in to the system with your ID and password.</p> <p>b. Compare the expected and realized PLOs and COs for the students for the time that has been entered.</p> <p><b>Department Head:</b></p> <p>a. Log in to the system with your ID and password.</p> <p>b. Compare the expected and</p>	<p><b>Computer/ Laptops:</b></p> <p>a. Used to sign into SPMS 4.0.</p> <p><b>Printer</b></p> <p>b. Used to print the attained PLO for the previous and current semesters on paper for comparison .</p>	<p><b>SPMS 4.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database or to save student data in the database</p>	<p><b>SPMS 4.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrators from the database.</p>	<b>Internet</b>	Used to sign to SPMS 4.0.

	<p>realized PLOs and COs for the students for the time that has been entered.</p> <p><b>Faculty:</b></p> <p>a. Log in to the system using your ID and password.</p> <p>b. Compare the expected and realized PLOs and COs for the students for the time that has been entered.</p> <p><b>Student:</b></p> <p>a. Log in to the system using your ID and password.</p> <p>b. Compare the expected and realized PLOs</p>				
--	--	--	--	--	--

	and COs for the students for the time that has been entered.				
<b>Department average of total PLO and CO achieved and attempted students</b>	<p><b>Dean/VC:</b></p> <p>a. Enter your ID to access the system and use your password.</p> <p>b. Choose the semester's desired viewing time range.</p> <p>c. See the overall PLO and CO average for the department as well as the number of students that attempted.</p> <p><b>Department Head:</b></p> <p>a. Log in to the system using</p>	<p><b>Computer/ Laptops:</b></p> <p>a. Used to sign into SPMS 4.0 Printer</p> <p>b. Used to print the attained PLO and CO for the previous and current semesters on paper for comparison .</p>	<p><b>SPMS 4.0:</b></p> <p>a. Used to create graphs of statistical analysis using data from the database or to save student data in the database</p>	<p><b>SPMS 4.0 database:</b></p> <p>a. All current information is kept here and can be updated by administrators from the database.</p>	<p><b>Internet</b></p> <p>Used to sign to SPMS 4.0.</p>

	<p>your ID and password.</p> <p>b. Choose the semester's desired viewing time range.</p> <p>c. See the overall PLO and CO average for the department as well as the number of students that attempted.</p> <p><b>Faculty:</b></p> <p>a. Enter your ID and password to log in.</p> <p>b. Check out the students' PLO and CO average across the entire department.</p> <p><b>Student:</b></p>				
--	---	--	--	--	--

	<p>a. Log in to the system with your ID and password.</p> <p>b. See the students' overall departmental average of PLO results.</p>				
<b>Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.</b>	<p><b>VC:</b></p> <p>1) Sign into the system using ID and Password.</p> <p>2) Select Student Enrollment Statistics tab and select Year and Semester under that tab</p> <p>3) View Student Enrollment Statistics of that Year and Semester.</p>	<b>Computer/ Laptops:</b>	<b>SPMS 4.0:</b>	<b>SPMS 4.0 database:</b>	<b>Internet</b>

	<p><b>Dean:</b></p> <p>1) Sign into the system using ID and Password.</p> <p>2) Select Student Enrollment Statistics tab and select Year and Semester under that tab</p> <p>3) View Student Enrollment Statistics of that Year and Semester.</p> <p><b>Department Head:</b></p> <p>1) Sign into the system using ID and Password.</p> <p>2) Select Student Enrollment Statistics tab and select Year</p>				
--	--	--	--	--	--

	and Semester under that tab  3) View Student Enrollment Statistics of that Year and Semester.					
--	---	--	--	--	--	--

## CHAPTER – 3 LOGICAL SYSTEM DIAGRAM

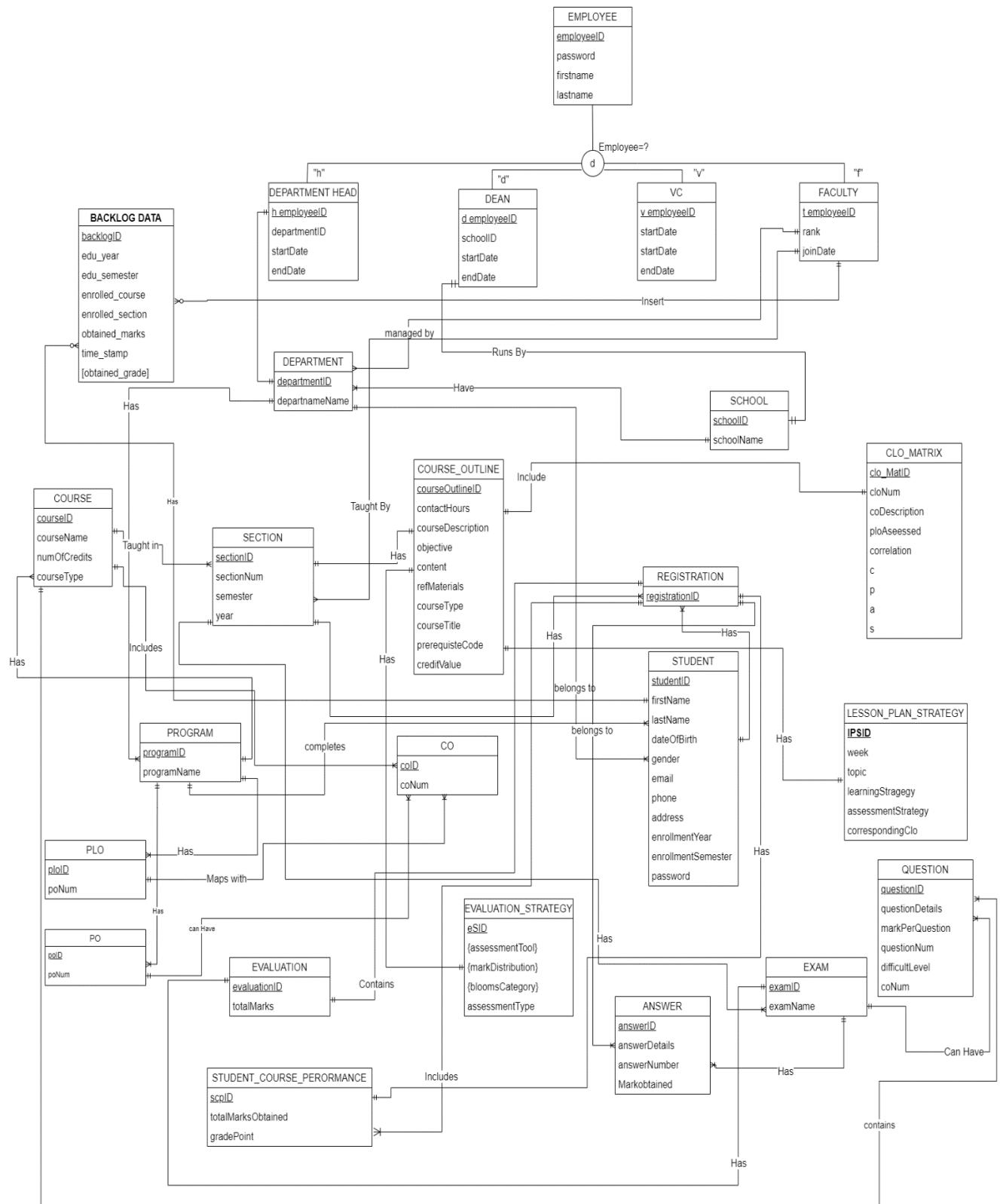
### I. BUSINESS RULES – PROPOSED SYSTEM

1. A student can only be in one department. A STUDENT has the following information: studentID, firstname, lastname, date of birth, gender, email, phone number, address, department ID, program ID, enrollmentYear, enrollmentSemester, and password. A department needs one or several Students.
2. A student may register for numerous events. SectionID, studentID, RegistrationID, enrolledCourse, educationalSemester, and educationalYear are all part of a registration. At least one student must complete a registration.
3. A part must require numerous registrations. There is at least one section in a registration. A section consists of the following: sectionID, sectionNum, courseID, facultyID, and year.
4. A registration could be a part of several EVALUATIONS. An evaluation must be associated with a single registration. EvaluationID, ExamID, RegistrationID, and TotalMarks are all included in an EVALUATION.

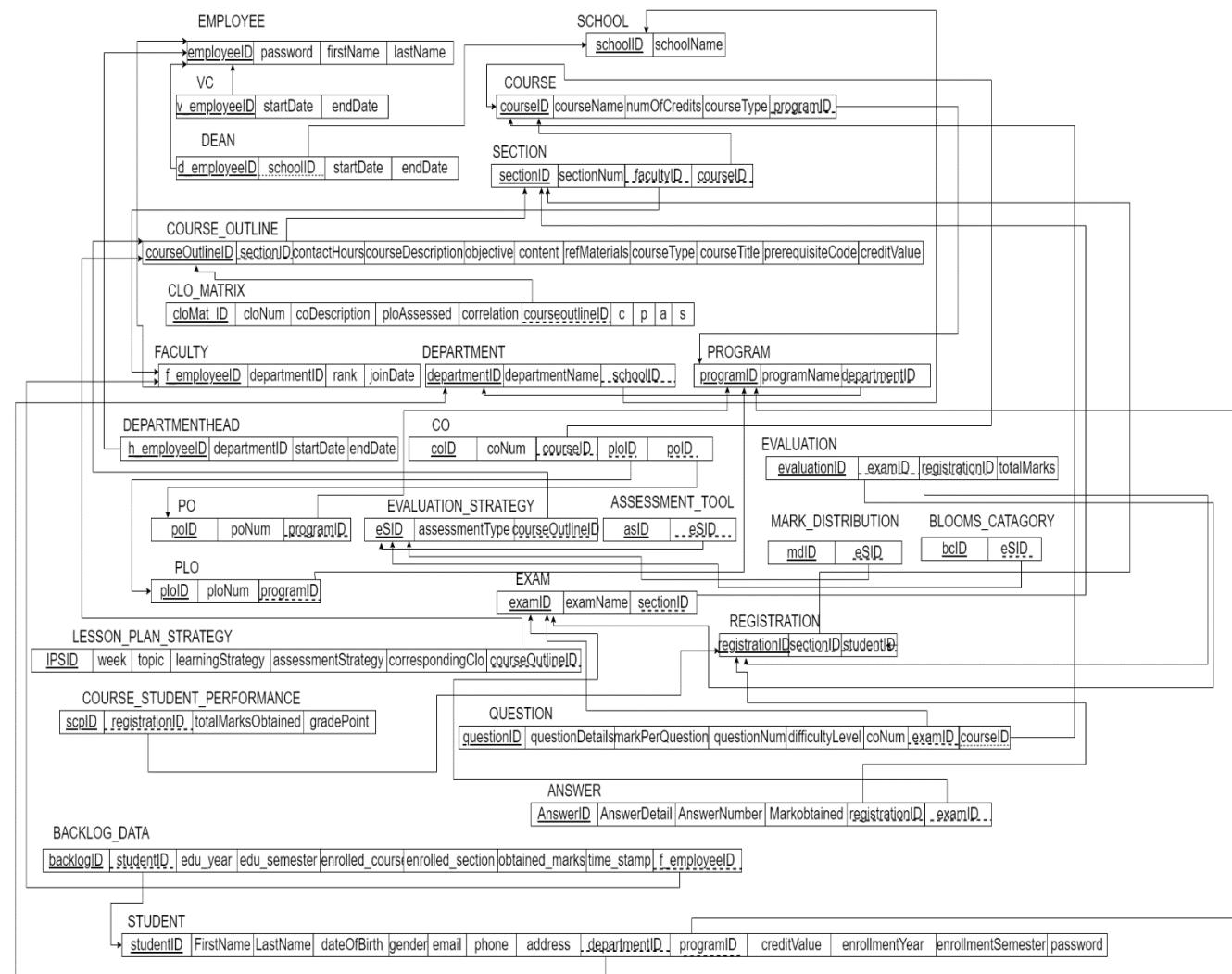
5. To map, a CO needs one PLO. A PLO must map with either one or several COs. PLO consists of programID, ploNum, and ploid.
6. There can be only one program in a PLO. One or more PLOs may be present in a program. Programs have departmentIDs, programIDs, and programNames. There must be one or more courses in a program. One course must make up a course.
7. Programs must be department-specific. One or more programs must be affiliated with a department. DepartmentID, DepartmentName, and SchoolID are contained in a DEPARTMENT.
8. Just one school may be found in a department. A SCHOOL must have a department or departments. A school's schoolID and schoolName are included.
9. There are four subtypes of employees (Dean, Department Head, Faculty, VC). Employee ID, password, first and last names are all part of an EMPLOYEE.
10. A school can only be managed by one person. A dean must oversee just one institution. A DEAN has a startDate, endDate, and schoolID.
11. There can only be one Department head per Department. A department head is only permitted to oversee one department. DepartmentID, startDate, and endDate are included in a DEPARTMENTHEAD.
12. There can only be one Department per Faculty. There must be one or more faculties in a department. DepartmentID, rank, and joinDate are all included in a FACULTY. A professor might oversee several divisions. A faculty member must teach one section only.
13. There is only one section in which a course outline fits. There can only be one course outline per part. The COURSE\_OUTLINE contains the course, sectionID, outlineID, and contact time, course Description, purpose, substance, and reference Materials, Course Type, Course Title, Prerequisite Code, and Credit Value

14. There must be precisely one CLO Matrix per course outline. A CLO matrix is specific to one course outline only. Here, clo\_MatID, cloNum, coDescription, ploAssessed, correlation, courseOutlineID, c,p,a, and s are all components of a CLO\_MATRIX.
15. There can only be one evaluation strategy per lesson plan strategy. There can only be one lesson plan strategy in an evaluation strategy. A LESSON\_PLAN\_STRATEGY contains the following: IPSID, week, topic, and learning. Strategy and evaluation courseOutlineID, correspondingClo, and strategy.
16. There is only one evaluation for a test. An exam evaluation is completed only once. A test only fits within one section. ExamID, ExamName, and SectionID are all parts of an EXAM. There must be one or more tests for a section.
17. There must be one or more questions on a test. Each question must be exclusive to one exam only. The following information included in a QUESTION are questionID, questionDetails, marksPerQuestion, questionNum, difficultyLevel, examID, and coNum. Answers to questions are given only once. There is only one question in a response.
18. A PO only has one program affiliation. There must be one or more PO in a program. PO consists of programID, poNum, and poID. A PO must be a member of one or more COs. A CO can only have one PO at a time.
19. For registration, a single student course performance evaluation is completed. Student course performance evaluations are completed exactly once for each registration. There is only one evaluation per registration. There is only one registration for an Evaluation.
20. For Backlog Data, Student and Faculty may have multiple Backlog Data but every Backlog Data must have exactly one Student and one Faculty.

## II. ENTITY RELATION DIAGRAM:



### III. ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL SCHEMA:



## IV. NORMALIZATION

<b>Entity</b>	<b>Attribute</b>	<b>Literal</b>
EMPLOYEE (e)	<u>employeeID</u>	<u>e1</u>
	firstName	e2
	lastName	e3
	password	e4
STUDENT (s)	<u>studentID</u>	<u>s1</u>
	firstName	s2
	lastName	s3
	dateOfBirth	s4
	gender	s5
	email	s6
	phone	s7
	address	s8
	departmentID	d1
	programID	r1
	enrollmentSemester	s9
SCHOOL (h)	enrollmentYear	s10
	password	s11
DEPARTMENT (d)	<u>schoolID</u>	<u>h1</u>
	schoolName	h2
DEPARTMENT (d)	<u>departmentID</u>	<u>d1</u>

	departmentName	d2
	schoolID	h1
PROGRAM (r)	<u>programID</u>	r1
	programName	r2
	departmentID	d1
	<u>f_employeeID</u>	<u>f1</u>
FACULTY (f)	departmentID	d1
	rank	f2
	joinDate	f3
	<u>h_employeeID</u>	<u>k1</u>
DEPARTMENTHEAD (k)	departmentID	d1
	startDate	k2
	endDate	k3
	<u>courseID</u>	<u>u1</u>
COURSE (u)	courseName	u2
	numOfCredits	u3
	courseType	u4
	programID	r1
	<u>sectionID</u>	<u>y1</u>
SECTION (y)	sectionNum	y2
	semester	y3
	courseID	u1
	facultyID	f1

	year	y4
COURSE_OUTLINE (c)	<u>courseOutlineID</u>	<u>c1</u>
	sectionID	y1
	contactHours	c2
	courseDescription	c3
	objective	c4
	content	c5
	refMaterials	c6
	courseType	c7
	courseTitle	c8
	prerequisiteCode	c9
LESSON_PLAN_STRATEGY (l)	creditValue	c10
	<u>ipsID</u>	<u>l1</u>
	week	l2
	topic	l3
	learningStrategy	l4
	assessmentStrategy	l5
	correspondingClo	l6
REGISTRATION (g)	courseOutlineID	c1
	<u>registrationID</u>	<u>g1</u>
	sectionID	y1
EXAM (i)	studentID	s1
	<u>examID</u>	<u>i1</u>

QUESTION (q)	examName	i2
	sectionID	y1
	<u>questionID</u>	<u>q1</u>
	questionDetails	q2
	markPerQuestion	q3
	questionNum	q4
	difficultyLevel	q5
	examID	i1
	courseID	u1
ANSWER (a)	coNum	q6
	<u>answerID</u>	<u>a1</u>
	answerDetails	a2
	answerNum	a3
	markObtained	a4
	registrationID	g1
EVALUATION_STRATEGY (t)	examID	i1
	<u>eSID</u>	<u>t1</u>
	assessmentTool	t2
	markDistribution	t3
	bloomsCategory	t4
EVALUTION (n)	courseOutlineID	c1
	<u>evaluationID</u>	<u>n1</u>
	examID	i1

	registrationID	g1
	totalMarks	n2
STUDENT_COURSE_PERFORMANCE (z)	<u>scpID</u>	<u>z1</u>
	registrationID	g1
	totalMarksObtained	z2
	gradePoint	z3
	obtainedGrade	z4
CLO_MATRIX (m)	<u>clo_MatID</u>	<u>m1</u>
	cloNum	m2
	coDescription	m3
	ploAssessed	m4
	correlation	m5
	courseOutlineID	c1
	c	m6
	p	m7
	a	m8
PLO (p)	s	m9
	<u>ploID</u>	<u>p1</u>
	ploNum	p2
CO (o)	programID	r1
	<u>coID</u>	<u>o1</u>
	coNum	o2
	courseID	u1

	ploID	p1
	poID	x1
PO (x)	<u>poID</u>	<u>x1</u>
	poNum	x2
	programID	r1
VC (v)	v <u>employeeID</u>	<u>v1</u>
	startDate	v2
	endDate	v3
DEAN (w)	<u>d_employeeID</u>	<u>w1</u>
	schoolID	h1
	startDate	w2
	endDate	w3
ASSESMENT_TOOL (as)	<u>asID</u>	<u>as1</u>
	eSID	t1
MARK_DISTRIBUTION (md)	<u>mdID</u>	<u>md1</u>
	eSID	t1
BLOOMS_CATEGORY (bc)	<u>bcID</u>	<u>bc1</u>
	eSID	t1
BACKLOG_DATA_T (bl)	<u>backlogID</u>	<u>bl1</u>
	edu_year	bl2
	edu_semester	bl3
	enrolled_course	bl4
	enrolled_section	bl5

	obtained_marks	bl6
	time_stamp	bl7
	facultyID	f1
	studentID	s1

### Functional Dependency Mapping Using Attribute:

employeeID →	firstName, lastName, password
studentID →	firstName, lastName, dateOfBirth, gender, email, phone, address, departmentID, programID, enrollmentSemester, enrollmentYear, password
schoolID →	schoolName
departmentID →	departmentName, schoolID
programID →	programName, departmentID
f_employeeID →	departmentID, rank, joinDate
h_employeeID →	departmentID. startDate, endDate
courseID →	courseName, numOfCredits, courseType, programID
sectionID →	sectionNum, semester, courseID, facultyID, year
courseOutlineID →	sectionID, contactHours, courseDescription, objective, content, refMaterials, courseType, courseTitle, prerequisiteCode, creditValue
ipsID →	week, topic, learningStrategy, assessmentStrategy, correspondingClo, courseOutlineID
registrationID →	sectionID, studentID
examID →	examName, sectionID

questionID →	questionDetails, markPerQuestion, questionNum, difficultyLevel, examID, courseID, coNum
answerID →	answerDetails, answerNum, markObtained, registrationID, examID
eSID →	assessmentTool, markDistribution, bloomsCategory, courseOutlineID
evaluationID →	examID, registrationID, totalMarks
scpID →	registrationID, totalMarksObtained, gradePoint, obtainedGrade
clo_MatID →	cloNum, coDescription, ploAssessed, correlation, courseOutlineID, c, p, a, s
ploID →	ploNum, programID
coID →	coNum, courseID, ploID, poID
poID →	poNum, programID
v_employeeID →	startDate, endDate
d_employeeID →	schoolID, startDate, endDate
asID →	eSID
mdID →	eSID
bcID →	eSID
backlogID →	edu_year
	edu_semester
	enrolled_course
	enrolled_section
	obtained_marks
	time_stamp
	facultyID

	studentID
--	-----------

### Functional Dependency Mapping Using Literal:

$e1 \rightarrow$	$e2, e3, e4$
$s1 \rightarrow$	$s2, s3, s4, s5, s6, s7, s8, d1, r1, s9, s10, s11$
$h1 \rightarrow$	$h2$
$d1 \rightarrow$	$d2, h1$
$r1 \rightarrow$	$r2, d1$
$f1 \rightarrow$	$d1, f2, f3$
$k1 \rightarrow$	$d1, k2, k3$
$u1 \rightarrow$	$u2, u3, u4, r1$
$y1 \rightarrow$	$y2, y3, u1, f1, y4$
$c1 \rightarrow$	$y1, c2, c3, c4, c5, c6, c7, c8, c9, c10$
$l1 \rightarrow$	$l2, l3, l4, l5, l6, c1$
$g1 \rightarrow$	$y1, s1$
$i1 \rightarrow$	$i2, y1$
$q1 \rightarrow$	$q2, q3, q4, q5, i1, u1, q6$
$a1 \rightarrow$	$a2, a3, a4, g1, i1$
$t1 \rightarrow$	$t2, t3, t4, c1$
$n1 \rightarrow$	$i1, g1, n2$
$z1 \rightarrow$	$g1, z2, z3, z4$
$m1 \rightarrow$	$m2, m3, m4, m5, c1, m6, m7, m8, m9$
$p1 \rightarrow$	$p2, r1$

o1 →	o2, u1, p1, x1
x1 →	x2, r1
v1 →	v2, v3
w1 →	h1, w2, w3
as1 →	t1
md1 →	t1
bc1 →	t1
bl1 →	bl2
	bl3
	bl4
	bl5
	bl6
	bl7
	f1
	s1

**1NF:** A relation that has a primary key and in which there are no repeating groups.

R1

<u>a1</u>	<u>k1</u>	<u>l1</u>	<u>m1</u>	<u>n1</u>	<u>o1</u>	<u>q1</u>	<u>t1</u>	<u>e1</u>	<u>z1</u>	a2	a3	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	d1	d2	e2	e3	e4
f1	f2	f3	g1	md1	bc1	as1	h1	h2	i1	i2	k2	k3	l2	l3	l4	l5	l6	m1	m2	m3	m4	m5	m6	m7	m8	n2
o2	p1	p2	q2	q3	q4	q5	q6	r1	r2	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	t2	t3	t4	u1	u2	u3
u4	v1	v2	v3	w1	w2	w3	x1	x2	y1	y2	y3	y4	z2	z3	z4	bl1	bl2	bl3	bl4	bl5	bl6	bl7				

**2NF:** A relation in first normal form in which every non-key attribute is fully functionally dependent on the primary key.

R10

<u>a1</u>	a2	a3	a4	g1	i1	y1	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	i2	y2	y3	u1	f1	y4	d1
m	d2	h1	h2	f1	f2	f3	r1	r2	u2	u3	u4	bl1	bl2	bl3	bl4	bl5	bl6	bl7						

R12

<u>k1</u>	k2	k3
-----------	----	----

R14

<u>m1</u>	m2	m3	m4	m5	m6	m7	m8
-----------	----	----	----	----	----	----	----

R15

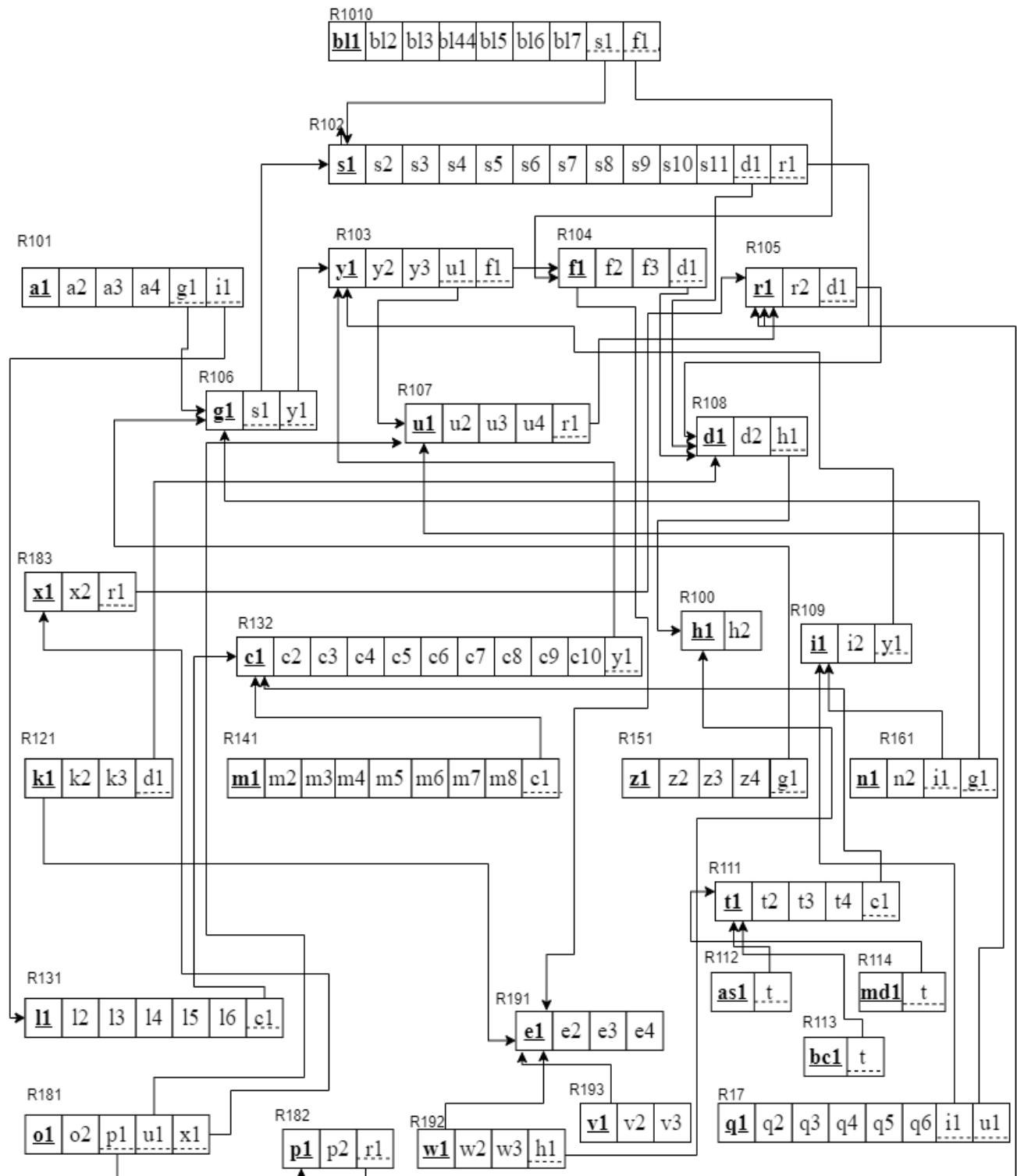
<u>z1</u>	z2	z3	z4
-----------	----	----	----

R16

<u>n1</u>	n2
-----------	----

PK

**3NF:** A relation that is in second normal form and has no transitive dependencies.



**BCNF:** All determinants are candidate keys. There is no determinant that is not a unique identifier. Here, all the relations already are in BCNF.

## V. DATA DICTIONARY

VC\_T

Name	Data Type	Size	Remark
v_employeeID	INTEGER	7	This is the foreign key from the Employee table. E.g: "4250"
startDate	DATE		This is starting date for the VC. E.g: "01-03-2020"
endDate	DATE		This is the date VC retire from his post. E.g: "01-03-2024"

STUDENT\_T

Name	Data Type	Size	Remark
studentID	INTEGER	7	This is the primary key for the Student table. E.g: "2030063".
firstName	VARCHAR	30	This is the first name of the student. E.g: "Roshni".
lastName	VARCHAR	30	This is the last name of the student. E.g:

			“Parvin”.
dateOfBirth	DATE		This is the birth date of the student. E.g: “30-11-1999”.
gender	VARCHAR	6	This is the gender of the student. E.g: “Female”.
email	VARCHAR	30	This is the email of the student. E.g: <a href="mailto:2030063@iub.edu.bd">“2030063@iub.edu.bd”</a>
phone	VARCHAR	11	This is the phone of the student. E.g: “01XXXXXXXXX”.
address	VARCHAR	50	This is the address of the student. E.g: “House 1,Road 4,Block D, Bashundhara RA
departmentID	VARCHAR	3	This is the foreign key from the Department table. E.g: “CSE”
programID	INTEGER	7	This is the foreign key from the Program table. E.g: “1”
enrollmentSemester	VARCHAR	10	This is the enrollment

			semester of the student.
enrollmentYear	VARCHAR	4	This is enrollment year of the student.

## STUDENT\_COURSE\_PERFORMANCE\_T

Name	Data Type	Size	Remark
scpID	INTEGER	7	This is the primarykey for this table
registrationID	INTEGER	7	This is the foreign key from registration table
totalMarksObtained	INTEGER	7	This is the total marks obtained by the student
gradePoint	FLOAT		This is the grade point achieved by the student

## SECTION\_T

Name	Data Type	Size	Remark

sectionID	INTEGER	7	This is the Primary Key for Section. E.g: "1"
sectionNum	INTEGER	11	This is the section number. E.g: "1"
semester	VARCHAR	6	This is the semester of the section. E.g: "Summer"
courseID	VARCHAR	6	This is the foreign key from the Course table. E.g: "CSE101"
facultyID	INTEGER	11	This is the foreign key from Faculty table. E.g: "1801"
year	YEAR	4	This is the year this section of this course was taken by this specific faculty

Name	Data Type	Size	Remark
schoolID	VARCHAR	7	This is the primary key of School. E.g: “SETS”
schoolName	VARCHAR	50	This is the name of the School. E.g: “School of Engineering, Technology & Science”.

## REGISTRATION\_T

Name	Data Type	Size	Remark
registrationID	INTEGER	7	This is the Primary Key for Registration.E.g: “0101010101”
sectionID	INTEGER	11	This is the foreign key from section table
studentID	INTEGER	7	This is the foreign key from studenttable

## QUESTION\_T

Name	Data Type	Size	Remark
questionID	INTEGER	11	This is the primarykey of this table
questionDetails	MEDIUMTEXT		This is the question
markPerQuestion	INTEGER	11	This is the mark each question contains
questionNum	INTEGER	11	This is the number of the question
difficultyLevel	INTEGER	11	This is the difficultylevel of the question
examID	VARCHAR	20	This is the foreign key from exam table
courseID	VARCHAR	6	This is the foreign key from course table
coNum	INTEGER	11	This is the CO number of the question

**PROGRAM\_T**

Name	Data Type	Size	Remark
programID	INTEGER	11	This is the primary key for a program. E.g: "1"
programName	VARCHAR	50	This is the name of the program. E.g: "Bachelor of Science"
departmentID	VARCHAR	3	This is the foreign key from the Department table. E.g: "CSE"

**PO\_T**

Name	Data Type	Size	Remark
poID	VARCHAR	5	This is the primarykey for Program Outcome. E.g: "PO1"
poNum	INTEGER	11	This is the PO number. E.g: "1"

programID	INTEGER	11	This is a foreign key from Program table. E.g: "1"
-----------	---------	----	---

## PLO\_T

Name	Data Type	Size	Remark
ploID	INTEGER	11	This is the primary key for Program Learning Outcome. E.g: "PLO1"
ploNum	INTEGER	11	This is the PLO number. E.g: "1"
programID	INTEGER	11	This is a foreign key from Program table. E.g: "1"

## LESSON\_PLAN\_STRATEGY\_T

Name	Data Type	Size	Remark
lpsID	INTEGER	11	This is the primarykey of the table
week	INTEGER	11	This is the week number

topic	MEDIUMTEXT		This is the topic name
learningStrategy	MEDIUMTEXT		This is the lesson plan strategy of that topic
assessmentStrategy	VARCHAR	10	This is the assessment strategy of that topic
courseOutlineID	INTEGER	11	This is the foreign key from course outline table

### FACULTY\_T

Name	Data Type	Size	Remark
f_employeeID	INTEGER	7	This is the foreign key from the Employee table. E.g: "4250"
departmentID	VARCHAR	3	This is the DepartmentID of the department faculty belongs to. E.g: "CSE"

rank	VARCHAR	30	This is the rank of the faculty. E.g: “Assistant Professor”
joinDate	DATE		This is starting date. E.g: “01-03-2020”

**EXAM\_T**

Name	Data Type	Size	Remark
examID	INTEGER	11	This is the primarykey for this table
examName	VARCHAR	30	This is the name of the exam
sectionID	INTEGER	11	This is the foreign key from exam table

**EVALUATION\_T**

Name	Data Type	Size	Remark
evaluationID	INTEGER	11	This is the primarykey for this table

examID	VARCHAR	20	This is the foreign key from examtable
registrationID	INTEGER	11	This is the foreign key from registration table
totalMarks	INTEGER	11	This is the total marks achieved by the student in a specific exam

#### EVALUATION\_STRATEGY\_T

Name	Data Type	Size	Remark
eSID	INTEGER	11	This is the primarykey for this table
courseOutlineID	INTEGER	11	This is the foreign key from course outline table

#### ASSESSMENT\_TOOL\_T

Name	Data Type	Size	Remark
asID	INTEGER	11	This is the primary key for this table
eSID	INTEGER	11	This is the foreign

			key from evaluation strategy table
--	--	--	------------------------------------

## MARK\_DISTRIBUTION\_T

Name	Data Type	Size	Remark
mdID	INTEGER	11	This is the primarykey for this table
eSID	INTEGER	11	This is the foreign key from evaluation strategy table

## BLOOMS\_CATEGORY\_T

Name	Data Type	Size	Remark
bcID	INTEGER	11	This is the primarykey for this table
eSID	INTEGER	11	This is the foreign key from evaluation strategytable

**EMPLOYEE\_T**

Name	Data Type	Size	Remark
employeeID	INTEGER	7	This is the primarykey for Employee table. E.g: "1801"
password	VARCHAR	10	This is the password of the employee
firstName	VARCHAR	50	This is the last name of the faculty. E.g: "Ahmed"
lastName	VARCHAR	50	This is the last name of the faculty. E.g: "Ahmed"

**DEPARTMENTHEAD\_T**

Name	Data Type	Size	Remark
h_employeeID	INTEGER	7	This is the foreign key from the Employee table.

			E.g: “4250”
departmentID	VARCHAR	3	This is the DepartmentID of the department HEAD manages. E.g: “CSE”
startDate	DATE		This is starting date. E.g: “01-03-2020”
endDate	DATE		This is the date HEAD retire from his post. E.g: “01-03-2024”

### DEPARTMENT\_T

Name	Data Type	Size	Remark
departmentID	VARCHAR	3	This is the primarykey for the Department table. E.g: “CSE”
departmentName	VARCHAR	50	This is the name of the department. E.g: “Computer Science and

			Engineering”.
schoolID	VARCHAR	5	This is a foreign key from the School table. E.g: “SETS”.

## DEAN\_T

Name	Data Type	Size	Remark
d_employeeID	INTEGER	7	This is the foreign key from the Employee table. E.g: “4250”
schoolID	VARCHAR	5	This is the SchoolID of the school DEAN manages. E.g: “SETS”
startDate	DATE		This is starting date. E.g: “01-03-2020”
endDate	DATE		This is the date DEAN retire from his post. E.g: “01-

			03-2024”
--	--	--	----------

**COURSE\_T**

Name	Data Type	Size	Remark
courseID	VARCHAR	6	This is the Primary Key for the Course. E.g: “CSE203”
courseName	VARCHAR	40	This is the name of the Course. E.g: “Discrete Mathematics”
numOfCredits	INTEGER	11	This is the number of credits for the Course. E.g: “3”
courseType	VARCHAR	10	This is the type of the Course. E.g: “Core”
programID	INTEGER	11	This is the foreign key from the program table. E.g: “1”

**COURSE\_OUTLINE\_T**

Name	Data Type	Size	Remark
courseOutlineID	INTEGER	11	This is the primarykey for this table
sectionID	INTEGER	11	This is the foreign key from the section table
courseDescription	MEDIUMTEXT		This is the description of the course
objective	MEDIUMTEXT		This is the objective of the course
content	MEDIUMTEXT		This is the content of the course
refMaterials	MEDIUMTEXT		This is the reference material
courseTitle	VARCHAR	1000	This is the title of the course
prerequisiteCode	VARCHAR	6	This is the prerequisite course code
creditValue	INTEGER	11	This is the credit value of the course

CO\_T

Name	Data Type	Size	Remark
coID	INTEGER	11	This is the primarykey for the CO table. E.g: “CO1”.
coNum	INTEGER	11	This is the CO number. E.g: 1,2 etc.
courseID	VARCHAR	6	This is the foreign key from the Course table. E.g: “CSE303”
ploID	VARCHAR	5	This is the foreign key from the PLO table. E.g: “PLO1”
poID	VARCHAR	6	This is the foreignkey from the PLOTable. E.g: “PO1”

CLO\_MATRIX\_T

Name	Data Type	Size	Remark
clo_MatID	INTEGER	11	This is the primarykey for this table
cloNum	INTEGER	11	This is the clo number
coDescription	MEDIUMTEXT		This is the co description
ploAssessed	VARCHAR	10	This is the name of the plo assessed
correlation	INTEGER	11	This is the correlation value or number
courseOutlineID	INTEGER	11	This is the foreign key from the course outline table
c	INTEGER	11	This is the bloom'scategory level
p	INTEGER	11	This is the bloom'scategory level
a	INTEGER	11	This is the bloom'scategory

			level
s	INTEGER	11	This is the bloom's category level

## ANSWER\_T

Name	Data Type	Size	Remark
answerID	INTEGER	11	This is the primarykey for this table
answerDetails	MEDIUMTEXT		This is the answer details
answerNum	INTEGER	11	This is the numberof the answer
markObtained	INTEGER	11	This is the mark obtained by the student for each answer
registrationID	INTEGER	11	This is the foreign key from registration table
examID	INTEGER	11	This is the foreign key from the exam

			table
--	--	--	-------

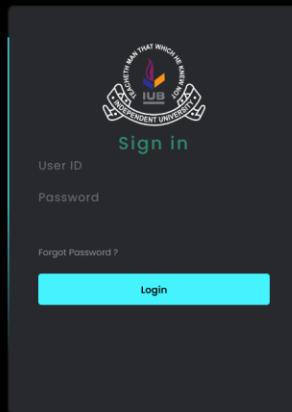
## BACKLOG\_DATA\_T

Name	Data Type	Size	Remark
<u>backlogID</u>	INTEGER	11	This is the primarykey for this table
edu_year	INTEGER	11	This is the educational year
edu_semester	CHAR	6	This is the educational semester
enrolled_course	CHAR	6	This is the enrolled course of the student
enrolled_section	INTEGER	2	This is the enrolled section of the student
obtained_marks	DECIMAL	(4,1)	This is the obtained marks of the student
time_stamp	TIMESTAMP		This is the time stamp upon submitting the backlog data by the

			faculty
facultyID	INTEGER	11	This is the foreign key from employee table
studentID	INTEGER	11	This is the foreign key from the student table

## CHAPTER-4 PHYSICAL SYSTEM DIAGRAM

### I. INPUT FORMS



```
<?php
include "connect.php" ;
session_start();
if (isset($_POST['id']) && isset($_POST['pass'])) {
    function validate($data){
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    $id = validate($_POST['id']);
    $pass = validate($_POST['pass']);
```

```
if ( strlen(strval($id)) === 4) {
    $sql = "SELECT * from employee_t where employeeID='$id' and password='$pass'";
    $result = mysqli_query($conn,$sql);
    $row = mysqli_fetch_assoc($result);
    if (mysqli_num_rows($result) === 1) {
        if (substr($id, 0, 1) === "1") {
            if ($row['employeeID'] === $id && $row['password'] === $pass) {
                $_SESSION['name'] = $row['firstName'] . " " . $row['lastName'];
                $_SESSION['id'] = $row['employeeID'];
                header("Location: H_Dashboard.php");
                exit();
            }
        } else {
            if ($row['employeeID'] === $id && $row['password'] === $pass) {
                $_SESSION['name'] = $row['firstName'] . " " . $row['lastName'];
                $_SESSION['id'] = $row['employeeID'];
                header("Location: F_Dashboard.php");
                exit();
            } }
        } else {
            header("Location: index.php?error = Incorrect ID or Password");
            exit();
        }
    } elseif ( strlen(strval($id)) === 7 ){

```

```
        }
    } elseif ( strlen(strval($id)) === 7 ) {
        $sql = "SELECT * from student_t where studentID='$id' and password='$pass'";
        $result = mysqli_query($conn,$sql);
        $row = mysqli_fetch_assoc($result);
        if (mysqli_num_rows($result) === 1) {
            if ($row['studentID'] === $id && $row['password'] === $pass) {
                //echo "hello " . $row['name'];
                $_SESSION['name'] = $row['firstName'] . " " . $row['lastName'];
                $_SESSION['id'] = $row['studentID'];
                header("Location: S_Dashboard.php");
                exit();
            }
        } else {
            header("Location: index.php?error = Incorrect ID or Password");
            exit();
        }
    } MD Tasin Rahman, 3 weeks ago • Add files via upload ...
} elseif ( strlen(strval($id)) === 5 ) {
    $sql = "SELECT * from authority_t where authorityID='$id' and password='$pass'";
    $result = mysqli_query($conn,$sql);
    $row = mysqli_fetch_assoc($result);
    if (mysqli_num_rows($result) === 1) {
        if ($row['authorityID'] === $id && $row['password'] === $pass) {
            //echo "hello " . $row['name'];
            $_SESSION['name'] = $row['firstName'] . " " . $row['lastName'];
            $_SESSION['id'] = $row['authorityID'];
            header("Location: H_Dashboard.php");
            exit();
        }
    } else {
        }
    } else {
        header("Location: index.php?error = Incorrect ID or Password");
        exit();
    }
} else {
    header("Location: index.php?error");
    exit();
}
}
```

Kazi  
Mubinul  
Hasan  
Shanto  
Faculty

Dashboard

PLO Analysis

Spider Chart Analysis

Data Entry

**Backlog Data**

Sign Out

STUDENT PERFORMANCE MONITORING SYSTEM

Semester:  
Autumn

Semester Year:  
2020

Enrolled Course:  
CSE101

Enrolled Section:

Student ID:

Obtained Marks:

Submit

```

if (isset($_POST['submit'])) {
    $studentID = $_POST['studentID'];
    $semester = $_POST['semester'];
    $year = $_POST['year'];
    $courseID = $_POST['courseID'];
    $section = $_POST['section'];
    $marks = $_POST['marks'];
    $facultyID = $_SESSION['id'];
    $timeStamp = date("Y-m-d H:i:s");

    $backlogQuery="INSERT INTO backlog_data_t (studentID, edu_year,
    edu_semester, enrolled_course, enrolled_section, obtained_marks,
    facultyID, time_stamp) VALUES
    ('$studentID', '$year', '$semester', '$courseID',
    '$section', '$marks', '$facultyID', '$timeStamp')";
    $backlogTable = mysqli_query($conn, $backlogQuery);

    $sectionQuery="INSERT INTO section_t (sectionNum, semester, courseID, facultyID, year) VALUES
    ('$section', '$semester', '$courseID','$facultyID', '$year')";
    $sectionTable = mysqli_query($conn, $sectionQuery);

    //Getting sectionID
    $result = mysqli_query($conn,
    "SELECT MAX(sectionID) AS secID
    FROM section_t");
    $row=mysqli_fetch_assoc($result);
    $secID=$row['secID'];
}

```

```
$registrationQuery="INSERT INTO registration_t (sectionID, studentID) VALUES
('{$secID}', '{$studentID}');
$registrationTable = mysqli_query($conn, $registrationQuery);

$examName="Backlog";
$examQuery="INSERT INTO exam_t (sectionID, examName) VALUES
('{$secID}', 'Backlog');
$examTable = mysqli_query($conn, $examQuery);

//Getting registrationID
$result = mysqli_query($conn,
"SELECT MAX(registrationID) AS regID
FROM registration_t");
$row=mysqli_fetch_assoc($result);
$regID=$row['regID'];
```

```
//student course performance
$gradePoint=0;
if( $marks >= 90 && $marks<=100)
    $gradePoint=4.0;
elseif( $marks>= 85 && $marks<=89)
    $gradePoint=3.7;
elseif($marks >= 80 && $marks<=84)
    $gradePoint=3.3;
elseif( $marks >= 75 && $marks<=79)
    $gradePoint=3.0;
elseif( $marks >= 70 && $marks <=74)
    $gradePoint=2.7;
elseif( $marks >= 60 && $marks <=69)
    $gradePoint=2.3;
elseif( $marks >= 65 && $marks <=64)
    $gradePoint=2.0;
elseif( $marks >= 55 && $marks <=59)
    $gradePoint=1.7;
elseif( $marks >= 50 && $marks <=54)
    $gradePoint=1.3;
elseif( $marks >= 45 && $marks<=49)
    $gradePoint=1.0;
elseif( $marks < 44 )
    $gradePoint=0.0;
$studCoursePerformanceQuery = "INSERT INTO student_course_performance_t(registrationID,
totalMarksObtained,gradePoint)
VALUES ('{$regID}', '{$marks}', '{$gradePoint}'')";
$studCoursePerformanceTable = mysqli_query($conn, $studCoursePerformanceQuery);
```

```

//Getting examID
$result = mysqli_query($conn,
"SELECT MAX(examID) AS examID
FROM exam_t");
$row=mysqli_fetch_assoc($result);
$examID=$row['examID'];

$ansMark = $marks/10;
$answerQuery="INSERT INTO answer_t (answerDetails, answerNum, markObtained,
registrationID,questionID, examID) VALUES
('Backlog', 1, '$ansMark', '$regID', 0, '$examID'),
('Backlog', 2, '$ansMark', '$regID', 0, '$examID'),
('Backlog', 3, '$ansMark', '$regID', 0, '$examID'),
('Backlog', 4, '$ansMark', '$regID', 0, '$examID')";
$answerTable = mysqli_query($conn, $answerQuery);

$questionQuery="INSERT INTO question_t (questionDetails, markPerQuestion, questionNum,
difficultyLevel, examID, courseID, coNum) VALUES
('Backlog', 10, 1, FLOOR(RAND()* (5-1+1))+1, '$examID', '$courseID', 1),
('Backlog', 10, 2, FLOOR(RAND()* (5-1+1))+1, '$examID', '$courseID', 2),
('Backlog', 10, 3, FLOOR(RAND()* (5-1+1))+1, '$examID', '$courseID', 3),
('Backlog', 10, 4, FLOOR(RAND()* (5-1+1))+1, '$examID', '$courseID', 4)";
$questionTable = mysqli_query($conn, $questionQuery);

```

```

//PO Table
$programID=0;
if($courseID=="CSE101"){
    $programID=13;}
elseif($courseID=="EEE131"){
    $programID=20;}
elseif($courseID=="ENG101"){
    $programID=9; }

$poQuery="INSERT INTO po_t (poNum, programID) VALUES
(FLOOR(RAND()* (12-1+1))+1, '$programID'),
(FLOOR(RAND()* (12-1+1))+1, '$programID'),
(FLOOR(RAND()* (12-1+1))+1, '$programID'),
(FLOOR(RAND()* (12-1+1))+1, '$programID')";
$poTable = mysqli_query($conn, $poQuery);

//Getting po/poID
$result = mysqli_query($conn,
"SELECT MAX(poID) AS poID
FROM po_t");
$row=mysqli_fetch_assoc($result);
$poID=$row['poID'];

```

```
//PLO Table :)  
$minPLO = $poID-3;  
$ploQuery="INSERT INTO plo_t (ploNum, programID)  
SELECT poNum, programID  
FROM po_t  
Where poID Between '$minPLO' AND '$poID'";  
$ploTable = mysqli_query($conn, $ploQuery);  
$ploID=$poID;  
  
//CO Table  
$coQuery="INSERT INTO co_t (coNum, courseID, ploID, poID) VALUES  
(1, '$courseID', '$ploID', '$poID'),  
(2, '$courseID', '$ploID', '$poID'),  
(3, '$courseID', '$ploID', '$poID'),  
(4, '$courseID', '$ploID', '$poID')";  
$coTable = mysqli_query($conn, $coQuery);  
}
```

STUDENT PERFORMANCE MONITORING SYSTEM

Kazi  
Mubinul  
Hasan  
Shanto  
Faculty

- Dashboard
- PLO Analysis
- Spider Chart Analysis
- Data Entry
- Backlog Data
- Sign Out

Choose File

No file chosen

Upload

Download Sample CSV file

View Backlog Data

Submit Backlog Data by importing CSV FILE

```

if (isset($_POST["upload"])) {
    if ($_FILES['fileToUpload']['name']) {
        $filename = explode(".", $_FILES['fileToUpload']['name']);
        if (end($filename) == "csv") {
            $handle = fopen($_FILES['fileToUpload']['tmp_name'], "r");
            $header = fgetcsv($handle);
            while ($data = fgetcsv($handle)) {
                $studentID = mysqli_real_escape_string($conn, $data[0]);
                $year = mysqli_real_escape_string($conn, $data[1]);
                $semester = mysqli_real_escape_string($conn, $data[2]);
                $courseID = mysqli_real_escape_string($conn, $data[3]);
                $section = mysqli_real_escape_string($conn, $data[4]);
                $marks = mysqli_real_escape_string($conn, $data[5]);
                $facultyID = $_SESSION['id'];
                $time = date("Y-m-d H:i:s");
                $query = "
                    INSERT INTO backlog_data_t (studentID, edu_year,
                    edu_semester, enrolled_course, enrolled_section, obtained_marks,
                    facultyID, time_stamp) VALUES
                    ('$studentID', '$year', '$semester', '$courseID',
                    '$section', '$marks', '$facultyID', '$time')
                ";
            }
        }
    }
}

```

```

mysqli_query($conn, $query);
$result = mysqli_query(
    $conn,
    "SELECT MAX(backlogID) AS backlogID
     FROM backlog_data_t"
);
$row = mysqli_fetch_assoc($result);
$backlogID = $row['backlogID'];
second ago • Uncommitted changes
$sectionQuery = "INSERT INTO section_t (sectionNum, semester, courseID, facultyID, year) VALUES
    ('$section', '$semester', '$courseID', '$facultyID', '$year')";
$sectionTable = mysqli_query($conn, $sectionQuery);

//Getting sectionID
$result = mysqli_query(
    $conn,
    "SELECT MAX(sectionID) AS secID
     FROM section_t"
);
$row = mysqli_fetch_assoc($result);
$secID = $row['secID'];

$registrationQuery = "INSERT INTO registration_t (sectionID, studentID) VALUES
    ('$secID', '$studentID')";
$registrationTable = mysqli_query($conn, $registrationQuery);

```

```
$examName = "Backlog";
$examQuery = "INSERT INTO exam_t (sectionID, examName) VALUES
    ('$secID', 'Backlog')";
$examTable = mysqli_query($conn, $examQuery);

//Getting registrationID
$result = mysqli_query(
    $conn,
    "SELECT MAX(registrationID) AS regID
        FROM registration_t"
);
$row = mysqli_fetch_assoc($result);
$regID = $row['regID'];
```

```
//student course performance
$gradePoint=0;
if( $marks >= 90 && $marks<=100)
    $gradePoint=4.0;
elseif( $marks>= 85 && $marks<=89)
    $gradePoint=3.7;
elseif($marks >= 80 && $marks<=84)
    $gradePoint=3.3;
elseif( $marks >= 75 && $marks<=79)
    $gradePoint=3.0;
elseif( $marks >= 70 && $marks <=74)
    $gradePoint=2.7;
elseif( $marks >= 60 && $marks <=69)
    $gradePoint=2.3;
elseif( $marks >= 65 && $marks <=64)
    $gradePoint=2.0;
elseif( $marks >= 55 && $marks <=59)
    $gradePoint=1.7;
elseif( $marks >= 50 && $marks <=54)
    $gradePoint=1.3;
elseif( $marks >= 45 && $marks<=49)
    $gradePoint=1.0;
elseif( $marks < 44 )
    $gradePoint=0.0;
$studCoursePerformanceQuery = "INSERT INTO student_course_performance_t(registrationID,
totalMarksObtained,gradePoint)
VALUES ('$regID', '$marks', '$gradePoint')";
$studCoursePerformanceTable = mysqli_query($conn, $studCoursePerformanceQuery);
```

```

//PO Table
$programID=0;
if($courseID=="CSE101"){
    $programID=13;
}
elseif($courseID=="EEE131"){
    $programID=20;
}
elseif($courseID=="ENG101"){
    $programID=9;
}

$poQuery="INSERT INTO po_t (poNum, programID) VALUES
(FLOOR(RAND()*(12-1+1))+1, '$programID'),
(FLOOR(RAND()*(12-1+1))+1, '$programID'),
(FLOOR(RAND()*(12-1+1))+1, '$programID'),
(FLOOR(RAND()*(12-1+1))+1, '$programID')";
$poTable = mysqli_query($conn, $poQuery);

//Getting po/ploID
$result = mysqli_query($conn,
"SELECT MAX(poID) AS poID
FROM po_t");
$row=mysqli_fetch_assoc($result);
$poID=$row['poID'];

```

```

//PLO Table :)
$minPLO = $poID - 3;
$ploQuery = "INSERT INTO plo_t (ploNum, programID)
    SELECT poNum, programID
    FROM po_t
    Where poID Between '$minPLO' AND '$poID'";
$ploTable = mysqli_query($conn, $ploQuery);
$ploID = $poID;

//CO Table
$coQuery = "INSERT INTO co_t (coNum, courseID, ploID, poID) VALUES
    (1, '$courseID', '$ploID', '$poID'),
    (2, '$courseID', '$ploID', '$poID'),
    (3, '$courseID', '$ploID', '$poID'),
    (4, '$courseID', '$ploID', '$poID')";
$coTable = mysqli_query($conn, $coQuery);

}
fclose($handle);
header("location: F_Import_CSV_File.php?updation=1");
} else {
    $message = '<label class="text-danger">Please Select CSV File Only</label>';
}
} else {
    $message = '<label class="text-danger">Please Select File</label>';
}
}
}

```

## II. OUTPUT FORMS

STUDENT PERFORMANCE MONITORING SYSTEM							
Student ID	Grade	Course	Section	Semester	Year	Faculty ID	Time Stamp
1531176	A-	CSE101	1	Autumn	2023	2483	2023-04-19 23:09:00
1811135	A	EEE131	2	Summer	2022	2483	2023-04-19 23:09:13
1711411	B-	ENG101	5	Autumn	2020	2259	2023-04-19 23:11:34
1711411	B-	ENG101	4	Summer	2022	2483	2023-04-19 23:32:07
1910876	B	CSE101	5	Summer	2022	2483	2023-04-19 23:32:33
1811135	A	CSE101	7	Spring	2021	2483	2023-04-19 23:33:07
1531176	A-	EEE131	4	Spring	2021	2483	2023-04-20 01:45:42
1531176	C+	CSE101	5	Summer	2023	2483	2023-04-20 10:10:36
1531176	A	CSE101	1	Autumn	2020	2483	2023-04-20 10:19:45
1531176	A	EEE131	1	Autumn	2023	2483	2023-04-20 10:22:44
1531176	C+	CSE101	2	Autumn	2020	2483	2023-04-20 10:23:35
1611001	B+	CSE101	13	Spring	2021	2483	2023-04-20 21:47:36
1711409	B-	CSE101	13	Spring	2022	2483	2023-04-20 21:47:36
1711411	A-	CSE101	13	Spring	2021	2483	2023-04-20 21:47:36
1720718	A-	CSE101	13	Spring	2023	2483	2023-04-20 21:47:36
1722021	A-	CSE101	13	Spring	2023	2483	2023-04-20 21:47:36
1810471	B	CSE101	13	Spring	2022	2483	2023-04-20 21:47:36
1811135	C-	CSE101	13	Spring	2020	2483	2023-04-20 21:47:36
1821772	F	CSE101	13	Spring	2023	2483	2023-04-20 21:47:36
1822089	C+	CSE101	13	Spring	2021	2483	2023-04-20 21:47:36
1531176	A	CSE101	4	Autumn	2020	2483	2023-04-20 22:58:00

```

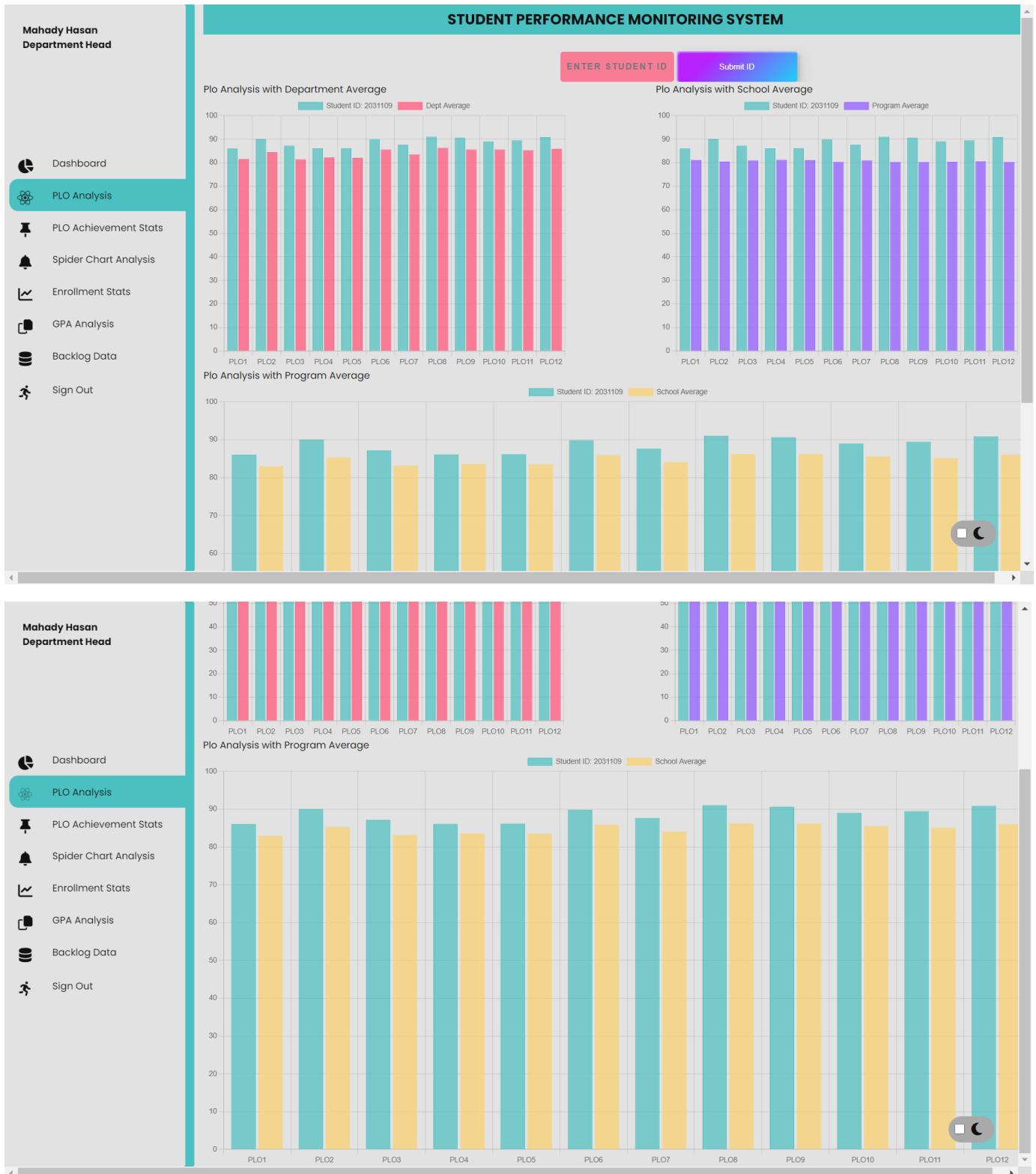
<?php
include 'connect.php';           You, 5 days ago • H_Backlog completed ...
$backlogData = "SELECT *
FROM backlog_data_t";
$result = mysqli_query($conn, $backlogData);
while ($row = mysqli_fetch_assoc($result)) {
?>
<?php
$grade = "Z";
if ($row['obtained_marks'] >= 90 && $row['obtained_marks'] <= 100)
    $grade = "A";
elseif ($row['obtained_marks'] >= 85 && $row['obtained_marks'] <= 89)
    $grade = "A-";
elseif ($row['obtained_marks'] >= 80 && $row['obtained_marks'] <= 84)
    $grade = "B+";
elseif ($row['obtained_marks'] >= 75 && $row['obtained_marks'] <= 79)
    $grade = "B";
elseif ($row['obtained_marks'] >= 70 && $row['obtained_marks'] <= 74)
    $grade = "B-";
elseif ($row['obtained_marks'] >= 60 && $row['obtained_marks'] <= 69)
    $grade = "C+";
elseif ($row['obtained_marks'] >= 65 && $row['obtained_marks'] <= 64)
    $grade = "C";
elseif ($row['obtained_marks'] >= 55 && $row['obtained_marks'] <= 59)
    $grade = "C-";
elseif ($row['obtained_marks'] >= 50 && $row['obtained_marks'] <= 54)
    $grade = "D+";
elseif ($row['obtained_marks'] >= 45 && $row['obtained_marks'] <= 49)
    $grade = "D";
elseif ($row['obtained_marks'] < 44)
    $grade = "F";
?>

```

```

<tr>
    <!-- <th scope="row">1</th> -->
    <!-- <td><?php echo $row['backlogID']; ?></td> -->
    <td><?php echo $row['studentID']; ?></td>
    <td><?php echo $grade; ?></td>
    <td><?php echo $row['enrolled_course']; ?></td>
    <td><?php echo $row['enrolled_section']; ?></td>
    <td><?php echo $row['edu_semester']; ?></td>
    <td><?php echo $row['edu_year']; ?></td>
    <td><?php echo $row['facultyID']; ?></td>
    <td><?php echo $row['time_stamp']; ?></td>
</tr>

```



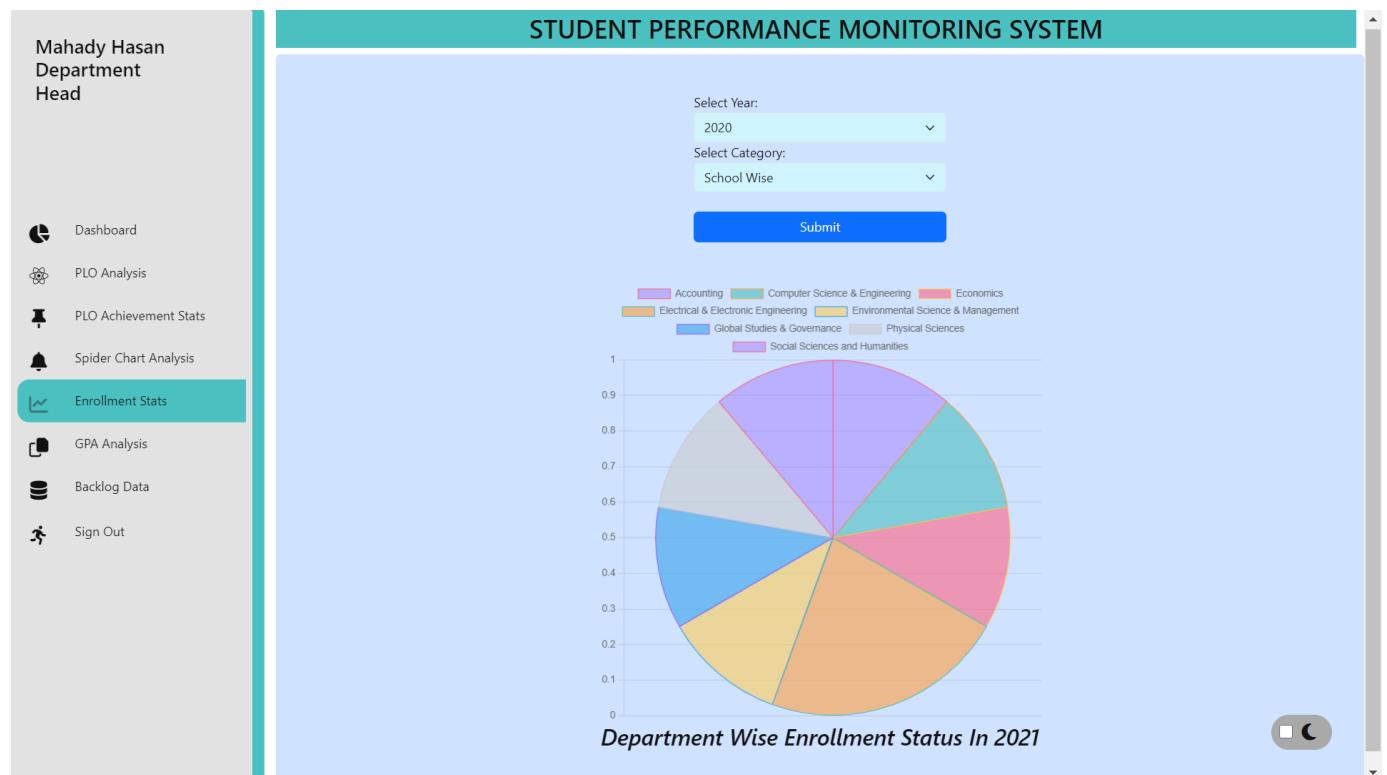
```
<?php  
if (isset($_POST['submit'])) {  
    $studentID = $_POST['studentID'];  
  
    include 'connect.php';  
  
    $sql = "SELECT plo.ploNum AS ploNum,  
            AVG((ans.markObtained/q.markPerQuestion)*100) AS percent  
        FROM registration_t AS r, answer_t AS ans, question_t AS q,  
             co_t AS co, plo_t AS plo  
       WHERE r.registrationID=ans.registrationID  
         AND ans.examID=q.examID  
         AND ans.answerNum=q.questionNum AND q.coNum=co.coNum  
         AND q.courseID=co.courseID AND co.ploID=plo.ploID  
         AND r.studentID='$studentID'  
        GROUP BY plo.ploNum,r.studentID";  
    $plo = mysqli_query($conn, $sql);  
  
    $sqlSch = "SELECT plo.ploNum AS ploNum,  
            AVG((ans.markObtained/q.markPerQuestion)*100) AS percent  
        FROM registration_t AS r, answer_t AS ans, question_t AS q,  
             co_t AS co, plo_t AS plo, student_t AS s, program_t AS p, department_t AS d  
       WHERE r.studentID=s.studentID  
         AND r.registrationID=ans.registrationID AND ans.examID=q.examID  
         AND ans.answerNum=q.questionNum  
         AND q.coNum=co.coNum AND q.courseID=co.courseID AND co.ploID=plo.ploID  
         AND s.departmentID=d.departmentID  
         AND d.schoolID=(SELECT d.schoolID FROM student_t AS s,  
                         department_t AS d WHERE s.studentID='$studentID'  
                         AND s.departmentID=d.departmentID)  
        GROUP BY plo.ploNum";  
    $resultSch = mysqli_query($conn, $sqlSch);
```

```

$sqlDept = "SELECT plo.ploNum AS ploNum, AVG((ans.markObtained/q.markPerQuestion)*100)
AS percent
FROM registration_t AS r, answer_t AS ans, question_t AS q,
co_t AS co, plo_t AS plo, student_t AS s WHERE r.studentID=s.studentID
AND r.registrationID=ans.registrationID AND ans.examID=q.examID
AND ans.answerNum=q.questionNum
AND q.coNum=co.coNum AND q.courseID=co.courseID AND co.ploID=plo.ploID
AND s.departmentID=(SELECT s.departmentID FROM student_t AS s
WHERE s.studentID='\$studentID')
GROUP BY plo.ploNum";
$resultDept = mysqli_query($conn, $sqlDept);

$sqlPro = "SELECT plo.ploNum AS ploNum,
AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
FROM registration_t AS r, answer_t AS ans, question_t AS q,
co_t AS co, plo_t AS plo, student_t AS s, program_t AS p
WHERE r.studentID=s.studentID
AND r.registrationID=ans.registrationID AND ans.examID=q.examID
AND ans.answerNum=q.questionNum
AND q.coNum=co.coNum AND q.courseID=co.courseID AND co.ploID=plo.ploID
AND s.programID=p.programID
AND s.programID=(SELECT s.programID FROM student_t AS s WHERE s.studentID='\$studentID')
GROUP BY plo.ploNum";
$resultPro = mysqli_query($conn, $sqlPro);
}

```



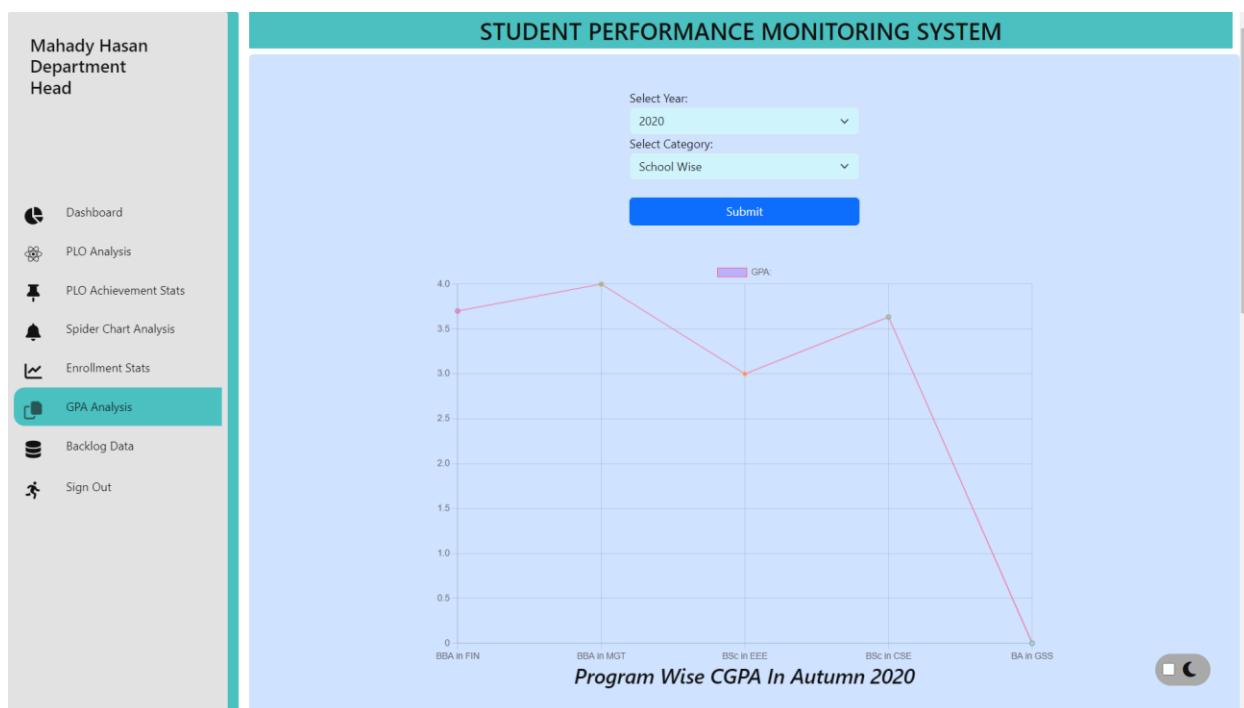
```

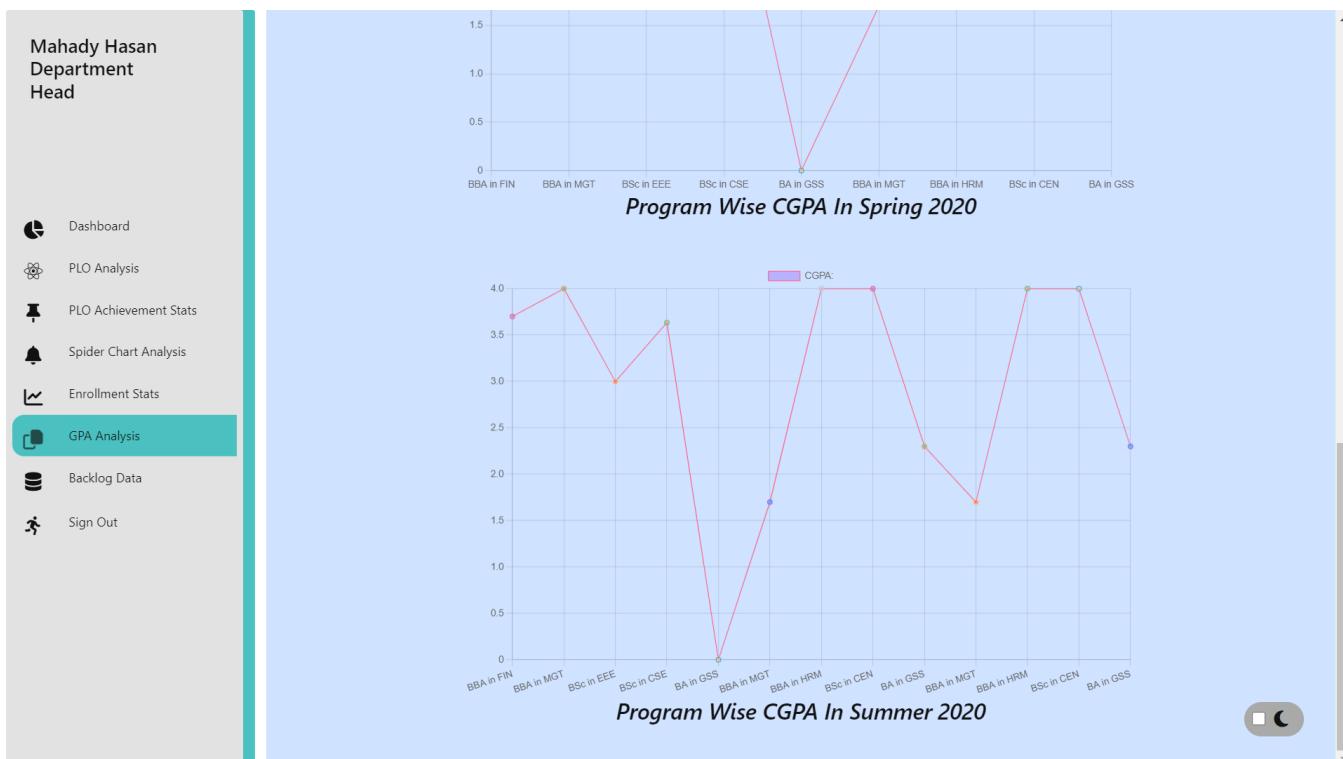
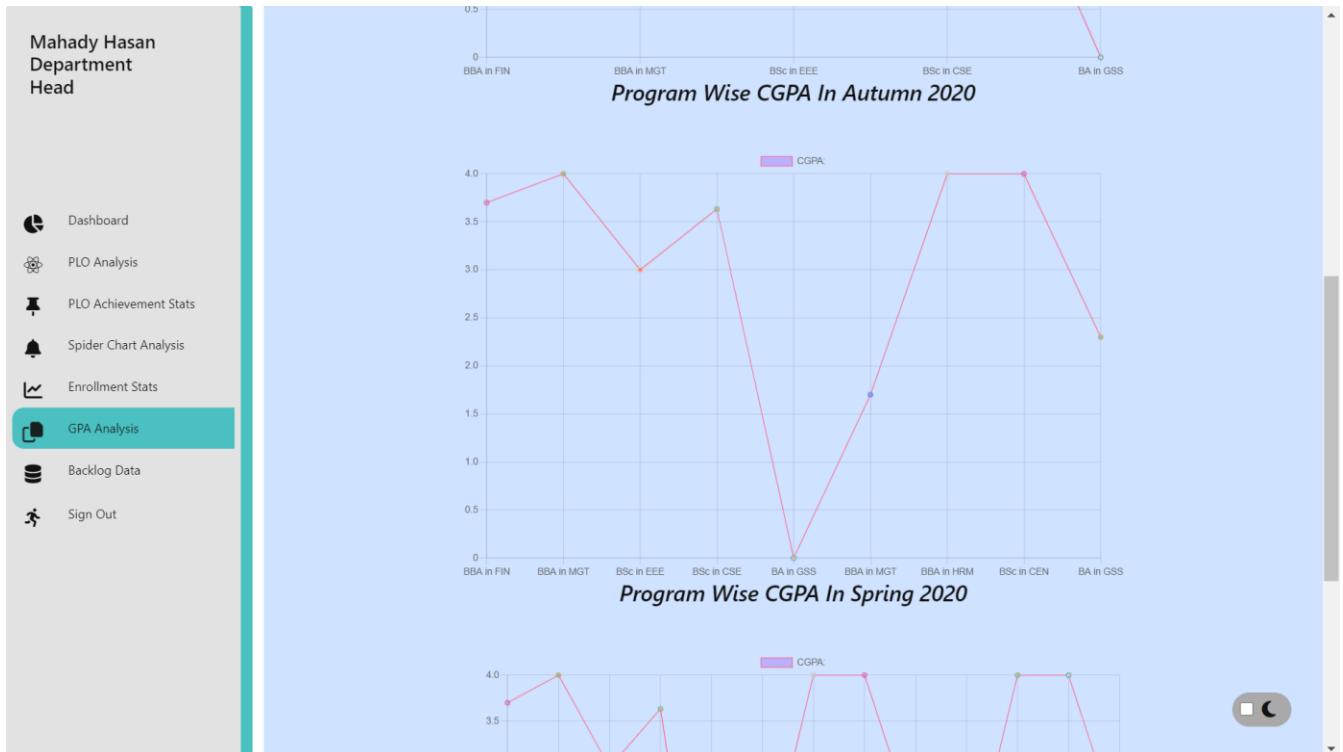
if (isset($_POST['submit'])) {
    $year = $_POST['year'];
    $category = $_POST['category'];
    if ($category == "School Wise") {
        $sql = "SELECT sch.schoolName as categoryName, COUNT(s.studentID) AS studNumber
                FROM student_t AS s INNER JOIN department_t AS d
                ON s.departmentID=d.departmentID
                INNER JOIN school_t AS sch
                ON d.schoolID=sch.schoolID
                WHERE d.departmentID=s.departmentID
                AND s.enrollmentYear='$year'
                GROUP BY sch.schoolName";

        $result = mysqli_query($conn, $sql);
    } elseif ($category == "Department Wise") {
        $sql = "SELECT d.departmentName AS categoryName, COUNT(s.studentID) AS studNumber
                FROM department_t AS d, student_t AS s
                WHERE s.enrollmentYear='$year' AND d.departmentID=s.departmentID
                GROUP BY s.departmentID";

        $result = mysqli_query($conn, $sql);
    } else {
        $sql = "SELECT p.programName AS categoryName,COUNT(s.studentID) AS studNumber
                FROM student_t AS s,program_t AS p
                WHERE s.enrollmentYear='$year' AND s.programID=p.programID
                GROUP BY p.programName";
    }
}

```





```
if (isset($_POST['submit'])) {
    $year = $_POST['year'];
    $category = $_POST['category'];
    if ($category == "School Wise") {
        $sqlAutumn = "SELECT sch.schoolID AS categoryName, AVG(scp.gradePoint) AS GPA
                      FROM student_t AS s, registration_t AS r,department_t AS d,school_t
                      AS sch,student_course_performance_t AS scp, section_t AS sec
                      WHERE scp.registrationID=r.registrationID AND r.studentID=s.studentID
                      AND r.sectionID=sec.sectionID AND s.departmentID=d.departmentID
                      AND d.schoolID=sch.schoolID AND (sec.semester='Autumn' OR sec.semester='autumn')
                      AND sec.year='$year'
                      GROUP BY sch.schoolID";
        $resultAutumn = mysqli_query($conn, $sqlAutumn);

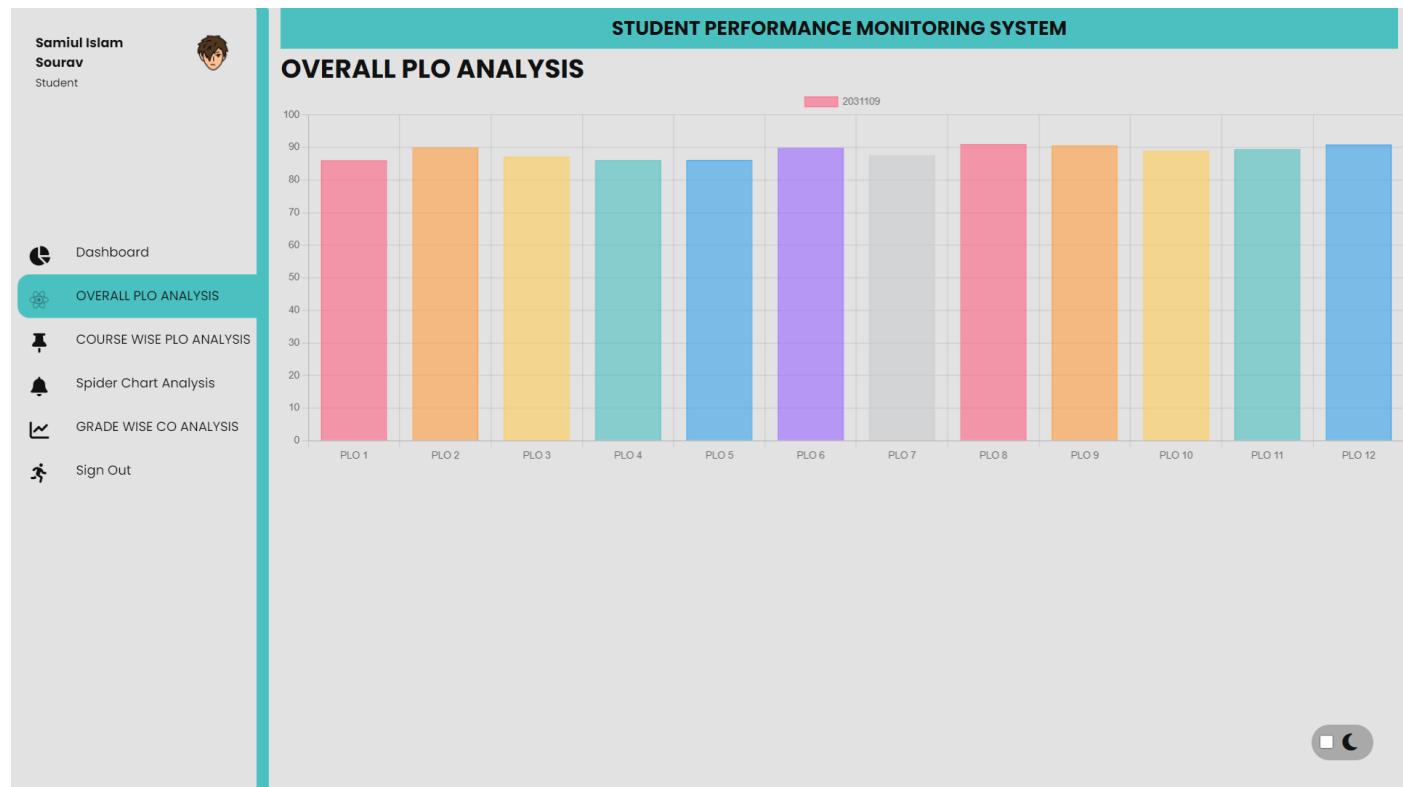
        $sqlSpring = "SELECT sch.schoolID AS categoryName, AVG(scp.gradePoint) AS GPA
                      FROM student_t AS s, registration_t AS r,department_t AS d,school_t
                      AS sch,student_course_performance_t AS scp, section_t AS sec
                      WHERE scp.registrationID=r.registrationID AND r.studentID=s.studentID
                      AND r.sectionID=sec.sectionID AND s.departmentID=d.departmentID
                      AND d.schoolID=sch.schoolID AND (sec.semester='Spring' OR sec.semester='spring')
                      AND sec.year='$year'
                      GROUP BY sch.schoolID";
        $resultSpring = mysqli_query($conn, $sqlSpring);

        $sqlSummer = "SELECT sch.schoolID AS categoryName, AVG(scp.gradePoint) AS GPA
                      FROM student_t AS s, registration_t AS r,department_t AS d,school_t
                      AS sch,student_course_performance_t AS scp, section_t AS sec
                      WHERE scp.registrationID=r.registrationID AND r.studentID=s.studentID
                      AND r.sectionID=sec.sectionID AND s.departmentID=d.departmentID
                      AND d.schoolID=sch.schoolID AND (sec.semester='Summer' OR sec.semester='summer')
                      AND sec.year='$year'
                      GROUP BY sch.schoolID";
        $resultSummer = mysqli_query($conn, $sqlSummer);
```

```
    } elseif ($category == "Department Wise") {  
        $sqlSummer = "SELECT s.departmentID AS categoryName, AVG(scp.gradePoint) as GPA  
                    FROM student_t AS s,student_course_performance_t  
                    AS scp, registration_t AS r, section_t AS sec  
                    WHERE r.registrationID=scp.registrationID AND  
                    r.studentID=s.studentID AND r.sectionID=sec.sectionID  
                    AND (sec.semester='summer' OR sec.semester='Summer') AND sec.year='$year'  
                    GROUP BY s.departmentID";  
        $resultSummer = mysqli_query($conn, $sqlSummer);  
  
        $sqlSpring = "SELECT s.departmentID AS categoryName, AVG(scp.gradePoint) as GPA  
                    FROM student_t AS s,student_course_performance_t  
                    AS scp, registration_t AS r, section_t AS sec  
                    WHERE r.registrationID=scp.registrationID AND  
                    r.studentID=s.studentID AND r.sectionID=sec.sectionID  
                    AND (sec.semester='spring' OR sec.semester='Spring') AND sec.year='$year'  
                    GROUP BY s.departmentID";  
        $resultSpring = mysqli_query($conn, $sqlSpring);  
  
        $sqlAutumn = "SELECT s.departmentID AS categoryName, AVG(scp.gradePoint) as GPA  
                    FROM student_t AS s,student_course_performance_t  
                    AS scp, registration_t AS r, section_t AS sec  
                    WHERE r.registrationID=scp.registrationID AND  
                    r.studentID=s.studentID AND r.sectionID=sec.sectionID  
                    AND (sec.semester='autumn' OR sec.semester='Autumn') AND sec.year='$year'  
                    GROUP BY s.departmentID";  
        $resultAutumn = mysqli_query($conn, $sqlAutumn);  
    } else {  
        $sqlSpring = "SELECT p.programName AS categoryName, AVG(scp.gradePoint) AS GPA  
                    FROM registration_t AS r, student_t AS s, student_course_performance_t  
                    AS scp, program_t AS p, section_t AS sec  
                    WHERE r.studentID=s.studentID AND scp.registrationID=r.registrationID  
                    AND r.sectionID=sec.sectionID AND s.programID=p.programID  
                    AND (sec.semester='spring' OR sec.semester='Spring') AND sec.year='$year'  
                    GROUP BY p.programID";  
        $resultSpring = mysqli_query($conn, $sqlSpring);  
    }  
}
```

```
$sqlSummer = "SELECT p.programName AS categoryName, AVG(scp.gradePoint) AS GPA
    FROM registration_t AS r, student_t AS s, student_course_performance_t
    AS scp, program_t AS p, section_t AS sec
    WHERE r.studentID=s.studentID AND scp.registrationID=r.registrationID
    AND r.sectionID=sec.sectionID AND s.programID=p.programID
    AND (sec.semester='summer' OR sec.semester= 'Summer')  AND sec.year='$year'
    GROUP BY p.programID";
$resultSummer = mysqli_query($conn, $sqlSummer);

$sqlAutumn = "SELECT p.programName AS categoryName, AVG(scp.gradePoint) AS GPA
    FROM registration_t AS r, student_t AS s, student_course_performance_t
    AS scp, program_t AS p, section_t AS sec
    WHERE r.studentID=s.studentID AND scp.registrationID=r.registrationID
    AND r.sectionID=sec.sectionID AND s.programID=p.programID
    AND (sec.semester='autumn' OR sec.semester= 'Autumn')  AND sec.year='$year'
    GROUP BY p.programID";
$resultAutumn = mysqli_query($conn, $sqlAutumn);
}
```



```

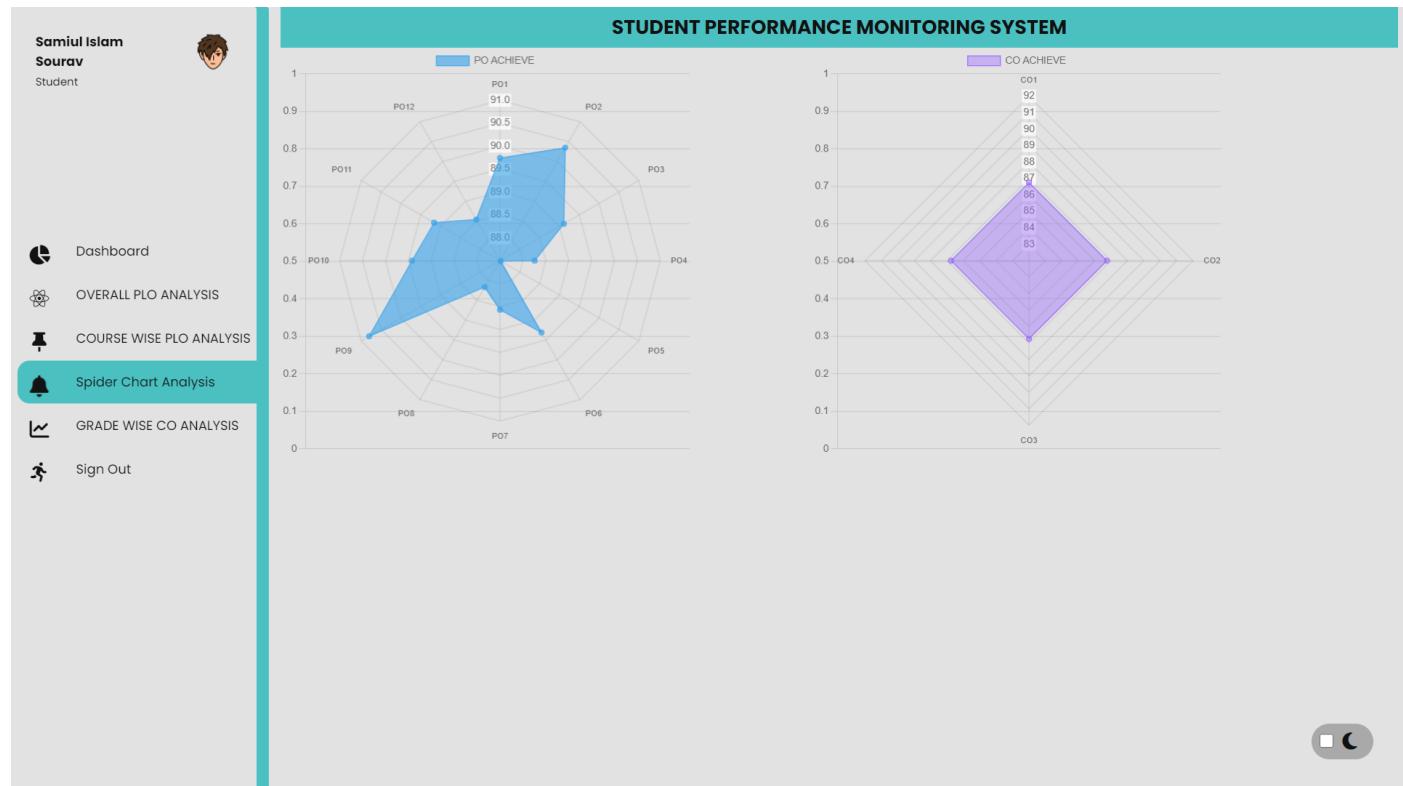
<?php
include 'connect.php';
session_start();
;
$studentID = $_SESSION['id'];

$sql = "SELECT plo.ploNum AS ploNum,
AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
FROM registration_t AS r, answer_t AS ans, question_t AS q,
co_t AS co, plo_t AS plo
WHERE r.registrationID=ans.registrationID
AND ans.examID=q.examID
AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
AND q.courseID=co.courseID AND co.ploID=plo.ploID
AND r.studentID='$studentID'
GROUP BY plo.ploNum,r.studentID";

$plo = mysqli_query($conn, $sql);

?>

```



```
SESSION_start();
$studentID = $_SESSION['id'];

$sql = "SELECT po.poNum AS poNum,
AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
FROM registration_t AS r, answer_t AS ans, question_t AS q,
co_t AS co, po_t AS po
WHERE r.registrationID=ans.registrationID
AND ans.examID=q.examID
AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
AND q.courseID=co.courseID AND co.poID=po.poID
AND r.studentID='$studentID'
GROUP BY po.poNum";

$spiderchart = mysqli_query($conn, $sql);

$sql1 = "SELECT q.coNum,
AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
FROM registration_t AS r, answer_t AS ans, question_t AS q,
co_t AS co, po_t AS po
WHERE r.registrationID=ans.registrationID      You, 1 second ago
AND ans.examID=q.examID
AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
AND r.studentID='$studentID'
GROUP BY q.coNum";
$covise = mysqli_query($conn, $sql1);
```

The screenshot shows the SPMS interface. On the left is a sidebar with a user profile for 'Samiul Islam Sourav' (Student) and a navigation menu with options like Dashboard, Overall PLO Analysis, Course Wise PLO Analysis, Spider Chart Analysis, Grade Wise CO Analysis (which is selected and highlighted in blue), and Sign Out. The main content area has a teal header 'STUDENT PERFORMANCE MONITORING SYSTEM' and a section titled 'GRADE WISE CO ANALYSIS'. It displays a bar chart for the enrolled course 'CSE101' with four bars labeled CO 1, CO 2, CO 3, and CO 4, each reaching approximately 90 on the y-axis scale from 0 to 100. Below the chart is a 'Submit' button.

```

if (isset($_POST['submit'])) {
    $studentID = $_SESSION["id"];
    $courseID = $_POST['courseID'];
    $gradeWiseCoQuery = "SELECT q.coNum,
        AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
        FROM registration_t AS r, answer_t AS ans, question_t AS q,
        co_t AS co, po_t AS po
        WHERE r.registrationID=ans.registrationID
        AND ans.examID=q.examID
        AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
        AND r.studentID= '$studentID'
        and q.courseID = '$courseID'
        Group by q.coNum";
    $gradeWiseCoTable = mysqli_query($conn, $gradeWiseCoQuery);
}

```

## CHAPTER – 5 CONCLUSION

### PROBLEM AND SOLUTION

#### I. Analysis Phase:

As there was no data available for the organizational activities during the analysis phase, one of the main issues was the confusion surrounding the Rich Picture and Six Element Analysis. Nonetheless, academic staff and other in order to clear up these misunderstandings, stakeholders were questioned, and data from the interview was gathered to provide a better knowledge of the system that was being designed.

#### II. Designing Phase:

During the Design Phase, some complications were encountered when constructing the EERD and Relational Schema, however they were easily resolved thanks to the faculty's regular feedback.

#### III. Implementation Phase:

All the System Requirements were completed successfully.

Front end developing tools: HTML, CSS, JavaScript

Backend developing tools: PHP

Database Integration: MySQL

#### **IV. Additional features and Future Development:**

The new features that will be added to the project are that the faculty now will be able to add the information of the student through google form directly. Besides, now both the PLO and CO will be mapped and which will also be analyzed and the statistics will also be shown with student evaluation.