TEAM NOOBSQUAD

PROJECT DATABASE MANAGEMENT SYSTEM

TEAM MEMBERS

Samiul Islam Sourav 2031109

2030063 Roshni Parvin

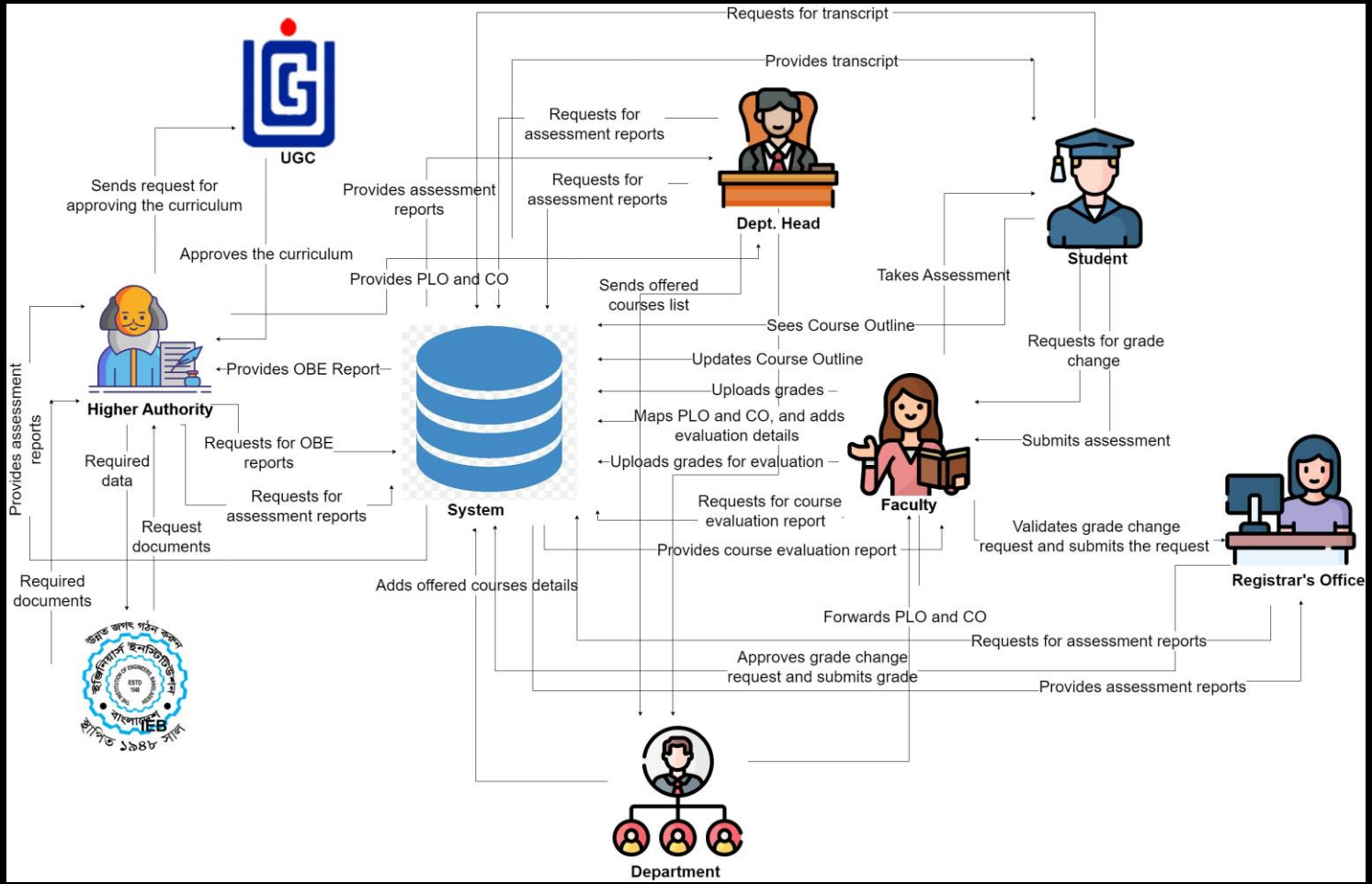
Md. Ezaz Ahamed 2021757

2031091 Md Tasin Rahman

Hasan Imam 2031121

2031254 Toufikul Islam Eyasin

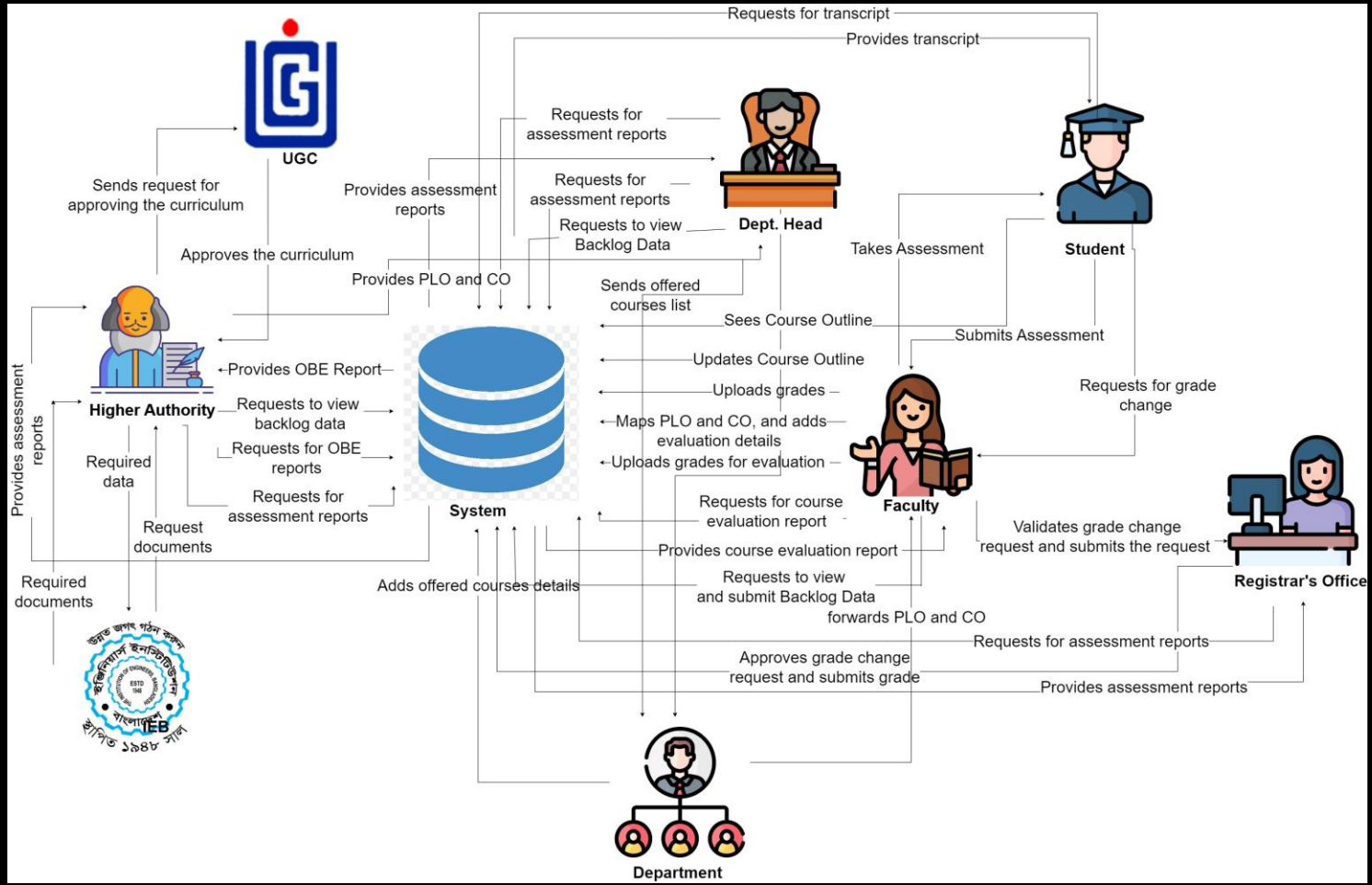
RICH PICTURE - Existing System



Problem ANALYSIS

Process Name	Stake Holder	Concerns (Problems)	Analysis (Reason of the problem)	Proposed Solution
Student Evaluation	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.
Grade Submission	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.
Backlog Data	Faculty: Higher Authority:	1.In the current system, there is no existence of the backlog data. 2. The lack of database makes it tough to manage the data of the students.	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.	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.

RICH PICTURE - Proposed System



SIX ELEMENT ANALYSIS - proposed system

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

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

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

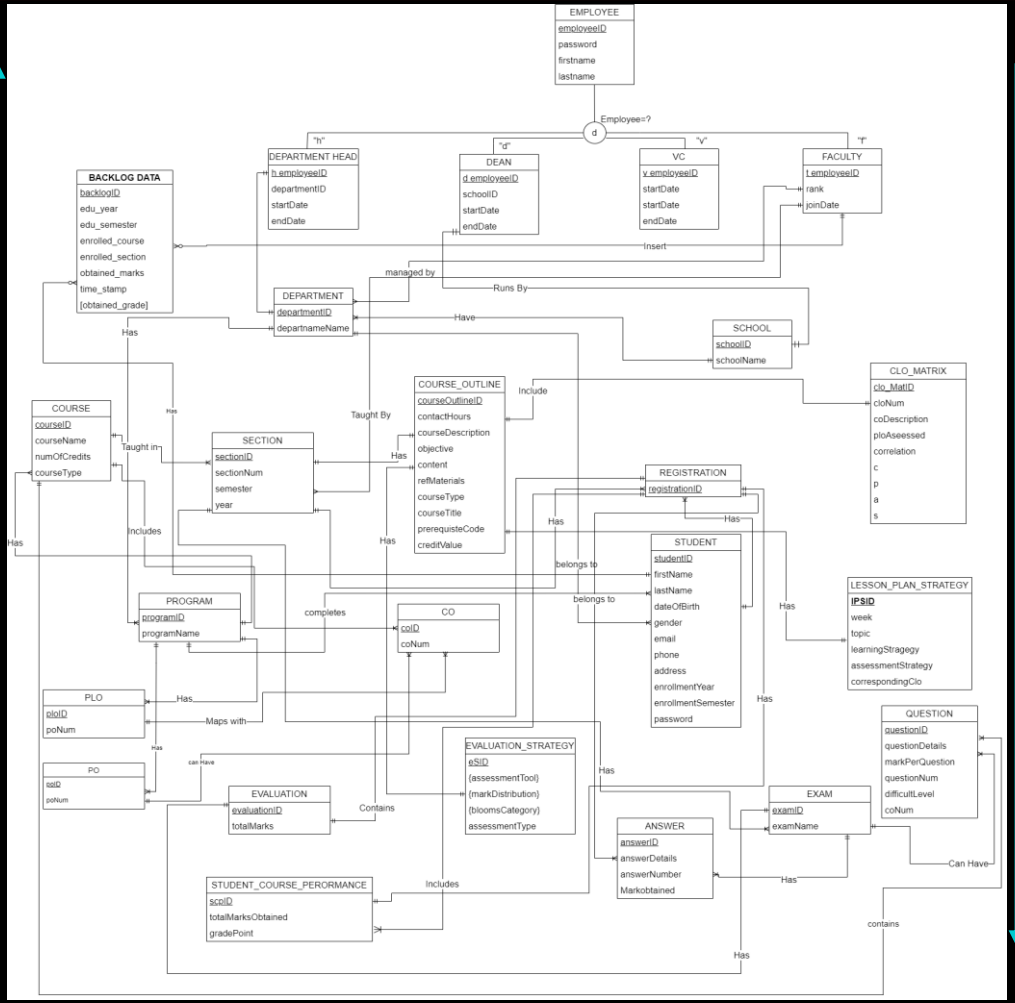
SIX ELEMENT ANALYSIS - proposed system

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

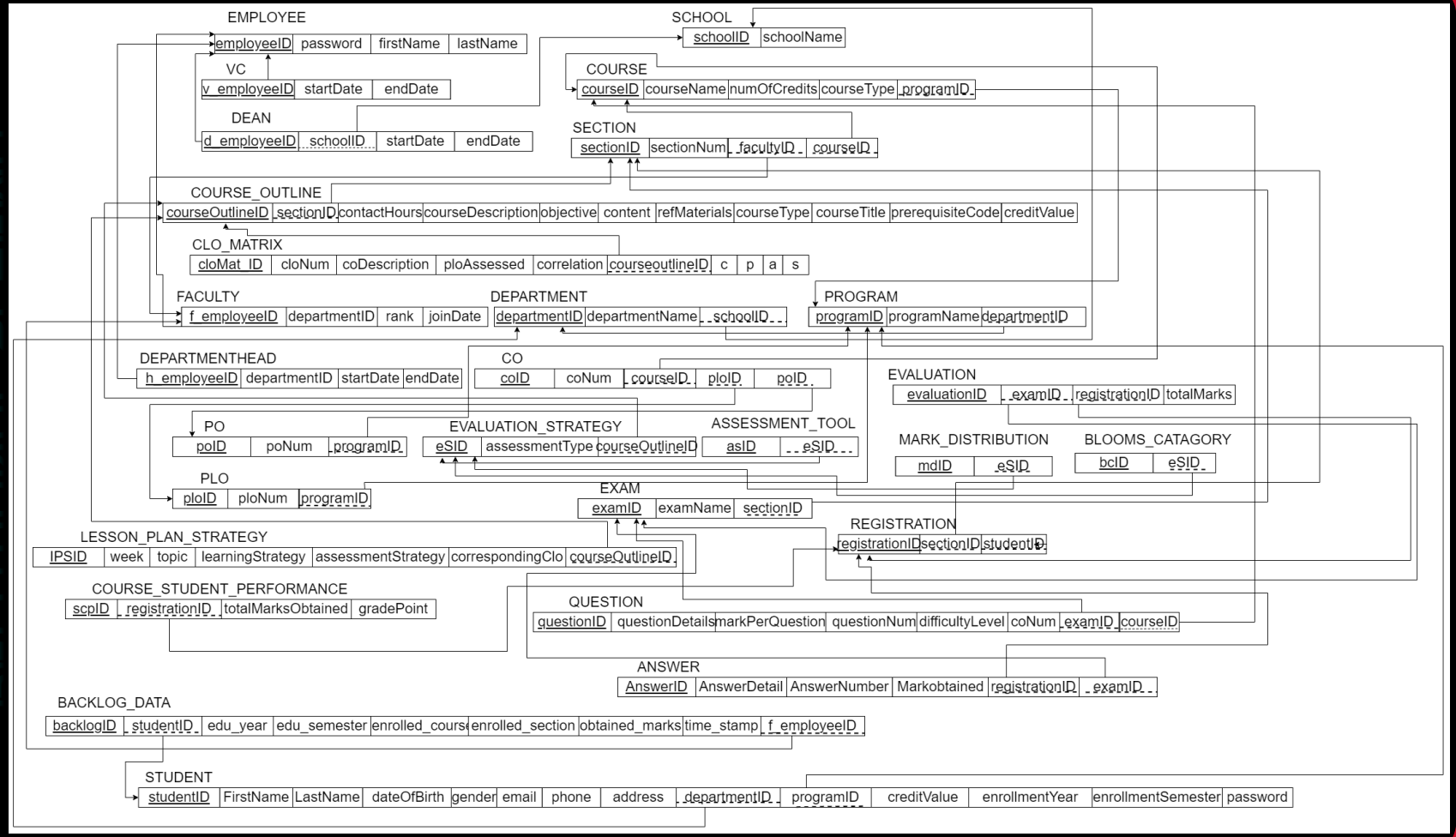
	and COs for the students for the time that has been entered.					
Department average of total PLO and CO achieved and attempted students	**Dean/VC:** a. Enter your ID to access the system and use your password. b. Choose the semester's desired viewing time range. c. See the overall PLO and CO average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

	a. Log in to the system with your ID and password. b. See the students' overall departmental average of PLO results.					
Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.	**VC:** 1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of that Year and Semester.		**Computer/ Laptops:** a. Used to sign into SPMS 4.0 Printer b. Used to print the attained statistics for the previous and current semesters on paper for comparison .	**SPMS 4.0:** a. Used to create graphs of statistical analysis using data from the database or to save student data in the database	**SPMS 4.0 database:** a. All current information is kept here and can be updated by administrators from the database.	**Internet** Used to sign to SPMS 4.0.

ERD


EMPLOYEE
employeeID
password
firstname
lastname
Employee=?
d
"h"
"d"
"v"
"f"
DEPARTMENT HEAD
h employeeID
departmentID
startDate
endDate
DEAN
d employeeID
schoolID
startDate
endDate
VC
v employeeID
startDate
startDate
endDate
FACULTY
f employeeID
rank
joinDate
BACKLOG DATA
backlogID
edu_year
edu_semester
enrolled_course
enrolled_section
obtained_marks
time_stamp
[obtained_grade]
Insert
managed by
Runs By
DEPARTMENT
departmentID
departnameName
Has
Have
SCHOOL
schoolID
schoolName
CLO_MATRIX
clo_MatID
cloNum
coDescription
ploAseessed
correlation
c
p
a
s
COURSE
courseID
courseName
numOfCredits
courseType
Taught in
Has
SECTION
sectionID
sectionNum
semester
year
Taught By
COURSE_OUTLINE
courseOutlineID
contactHours
courseDescription
objective
content
reffMaterials
courseType
courseTitle
prerequisteCode
creditValue
Include
Has
Includes
REGISTRATION
registrationID
Has
STUDENT
studentID
firstName
lastName
dateOfBirth
gender
email
phone
address
enrollmentYear
enrollmentSemester
password
belongs to
LESSON_PLAN_STRATEGY
IPSID
week
topic
learningStragegy
assessmentStrategy
correspondingClo
PROGRAM
programID
programName
completes
CO
coID
coNum
PLO
ploID
poNum
Has
Maps with
PO
poID
poNum
can Have
EVALUATION
evaluationID
totalMarks
Contains
EVALUATION_STRATEGY
eSID
{assessmentTool}
{markDistribution}
{bloomsCategory}
assessmentType
QUESTION
questionID
questionDetails
markPerQuestion
questionNum
difficultLevel
coNum
EXAM
examID
examName
Can Have
ANSWER
answerID
answerDetails
answerNumber
Markobtained
STUDENT_COURSE_PERORMANCE
scpID
totalMarksObtained
gradePoint
Includes
contains

RELATIONAL SCHEMA



Funtional Dependency

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, coursed, facultyID, year
courseOutlineID →	sectionID, contactHours, courseDescription, objective, content, refMaterials, courseType, courseTitle, prerequsiteCode, creditValue
ipsID →	week, topic, learningStrategy, assessmentStrategy, correspondingClo, courseOutlineID
registrationID →	sectionID, studentID
examID →	examName, sectionID
questionID →	questionDetails, markPerQuestion, questionNum, difficultyLevel, examID, coursed, 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
backlog_courseID →	backlogID
	courseID
backlog_sectionID →	backlogID
	sectionID

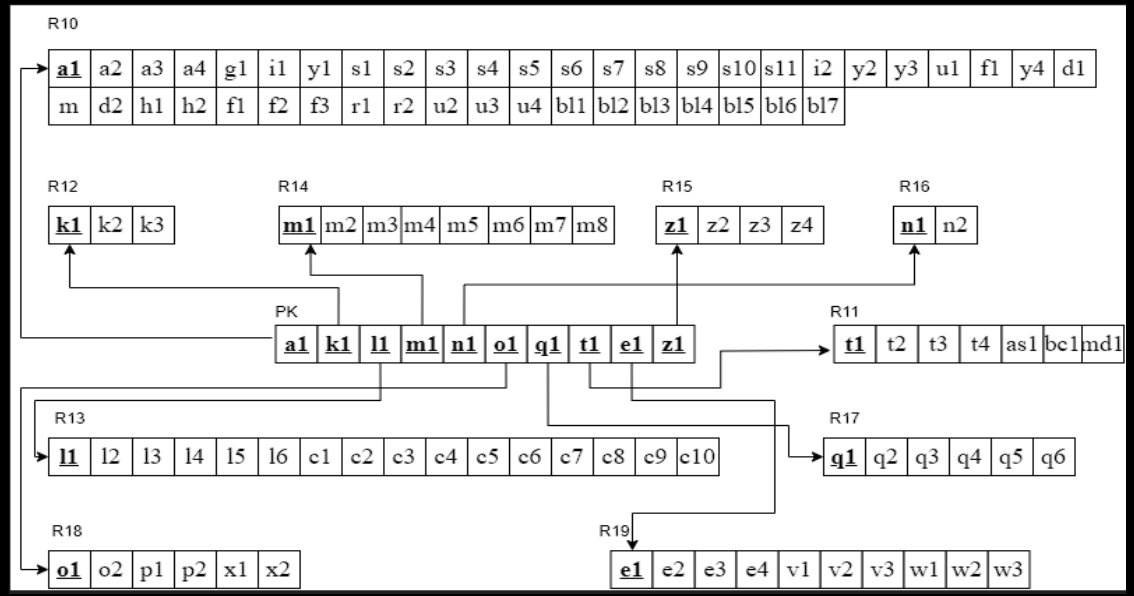
NORMALIZATION

1NF:

R1

a1	**k1**	**l1**	**m1**	**n1**	**o1**	**q1**	**t1**	**e1**	**z1**	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				

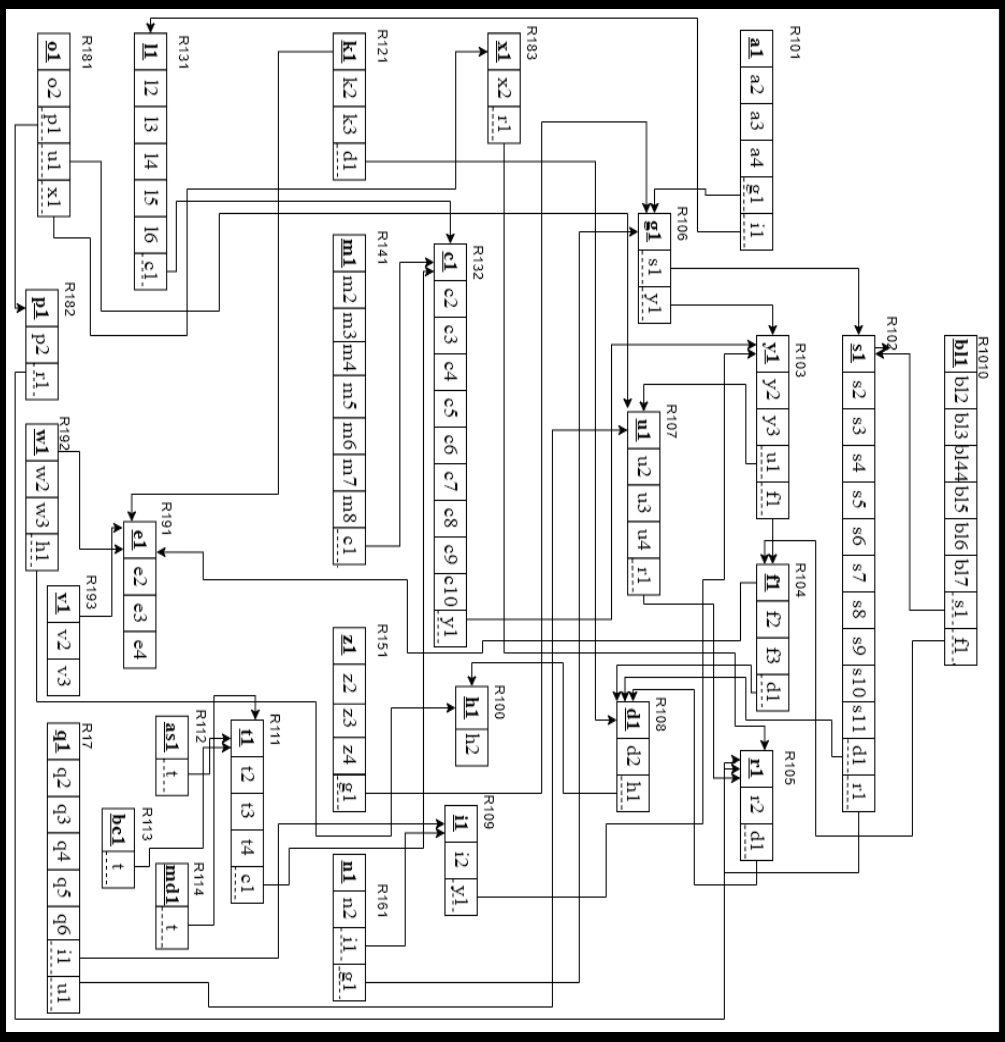
2NF:



NORMALIZATION

- continued

3NF:



DATA DICTIONARY

V. DATA DICTIONARY

VC_T

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

STUDENT_T

Name	Data Type	Size	Remark
studentID	INTEGER	7	This is the primary keyfor 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 ofthe 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: "2031254@iub.edu.bd"
phone	VARCHAR	11	This is the phone of the student. E.g: "01XXXXXXXX".
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 thestudent.
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

DATA DICTIONARY – continued

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

SCHOOL_T

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 ofEngineering, 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

DATA DICTIONARY – continued

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"

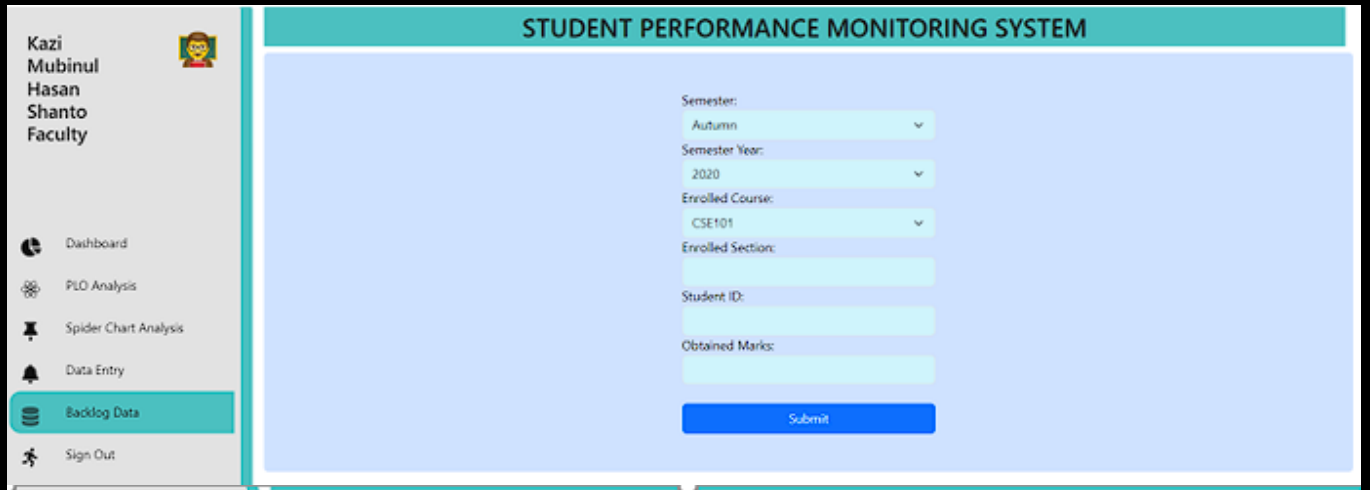
LESSON_PLAN_STRATEGY_T

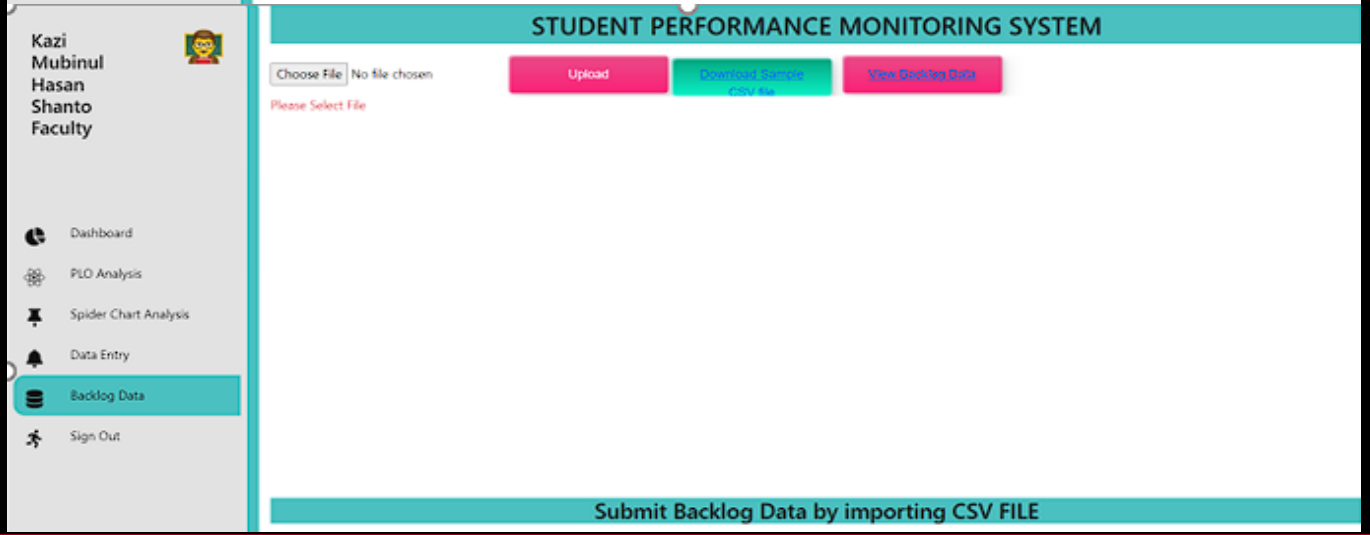
Name	Data Type	Size	Remark
lpsID	INTEGER	11	This is the primarykey of the table
week	INTEGER	11	This is the week number
topic	MEDIUMTEXT		This is the topic name
learningStrategy	MEDIUMTEXT		This is the lesson plan strategy of that topic
assessmentStrategy	VARCHAR	10	This is the assessment strategy of 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"

INPUT FORMS


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


Kazi Mubinul Hasan Shanto
Faculty
Dashboard
PLO Analysis
Spider Chart Analysis
Data Entry
Backlog Data
Sign Out
STUDENT PERFORMANCE MONITORING SYSTEM
Choose File
No file chosen
Upload
Download Sample CSV file
View Backlog Data
Please Select File
Submit Backlog Data by importing CSV FILE

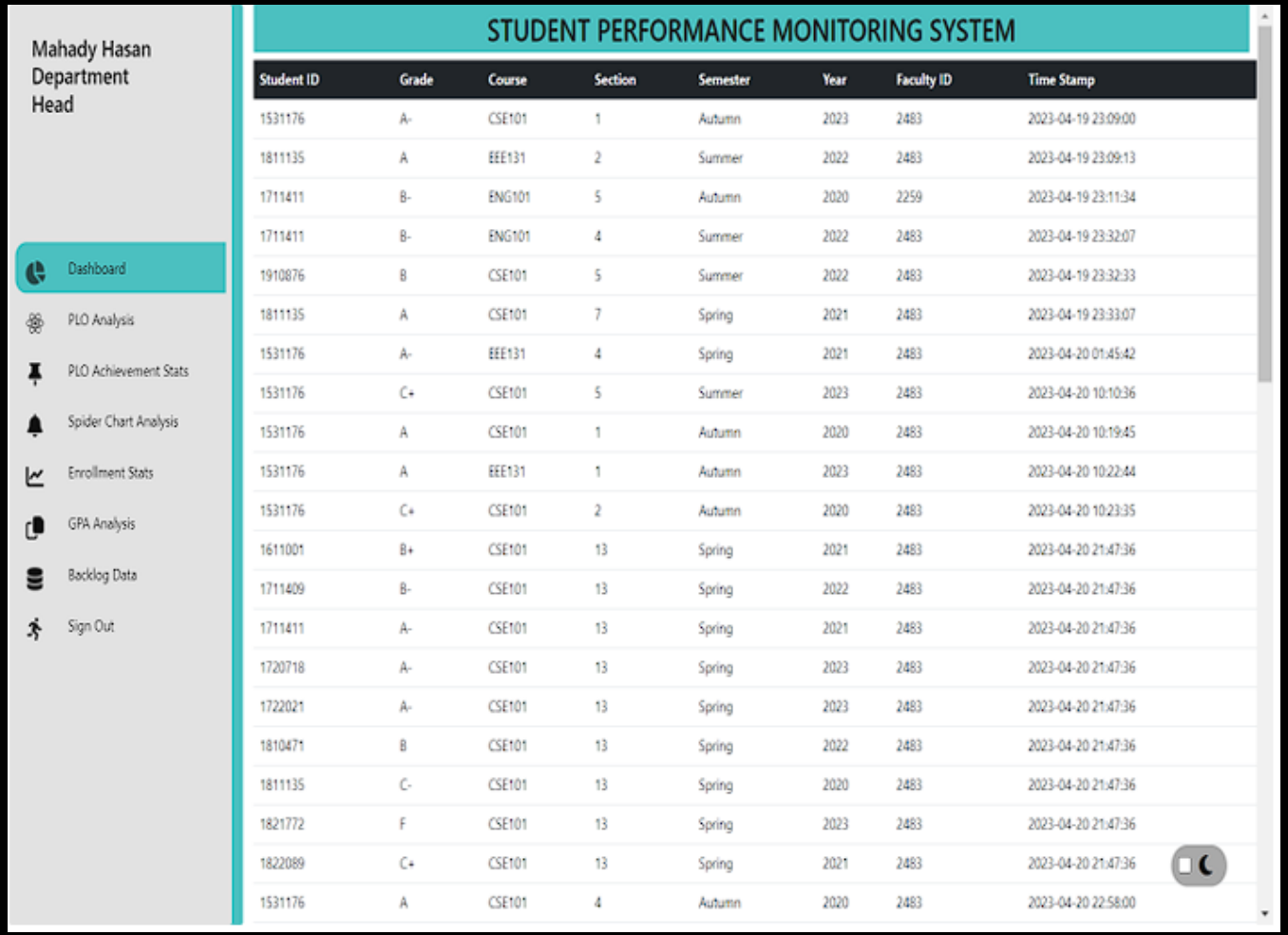
CODE OF INPUT FORMS

```php
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);
```

CODE OF INPUT FORMS

```php
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')

                    ";
```

OUTPUT FORMS



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

CODE OF OUTPUT FORMS

```php
<?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";
    ?>
```

```php
<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>
```

THANK YOU!