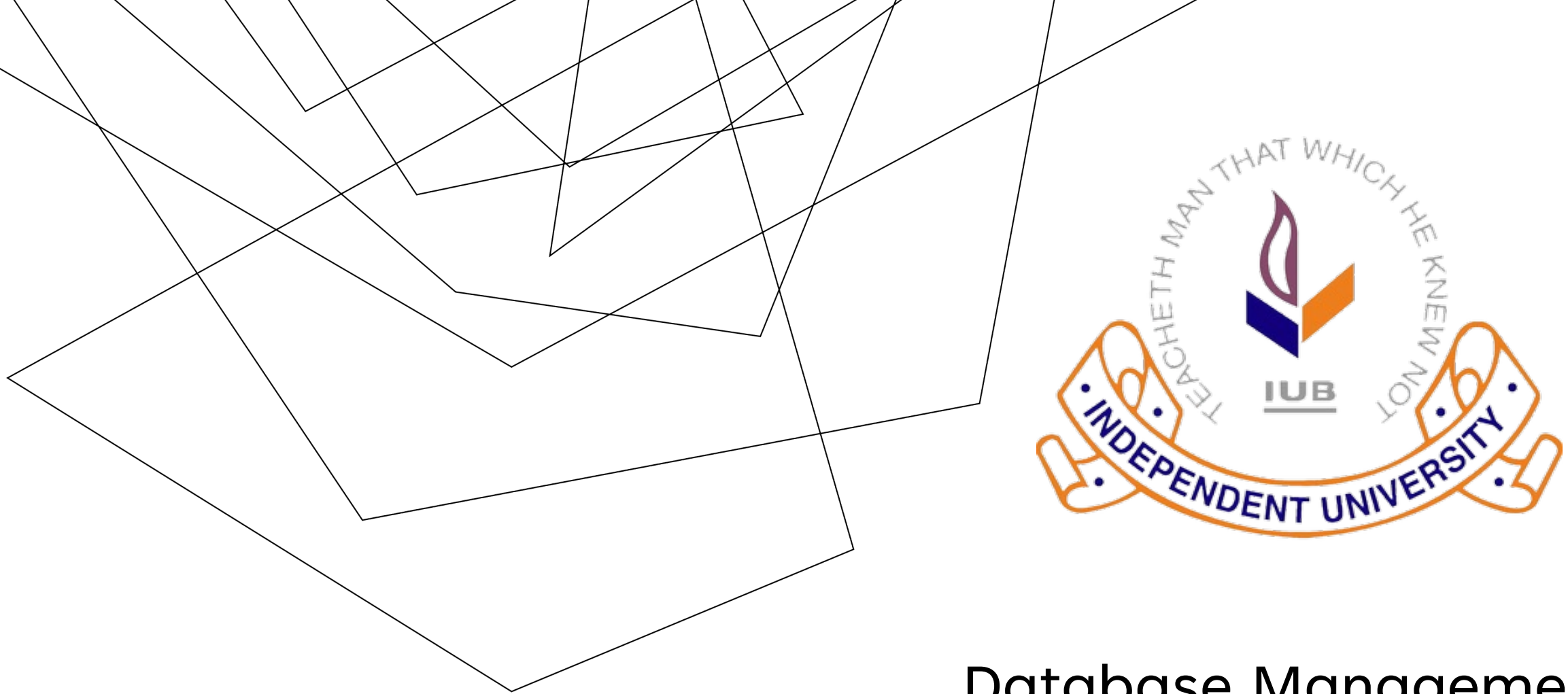


Database Management Project Presentation

Group - 18
Team Database

Shahriar Amin Ronok – 2031361

Tushar Basak – 2022315

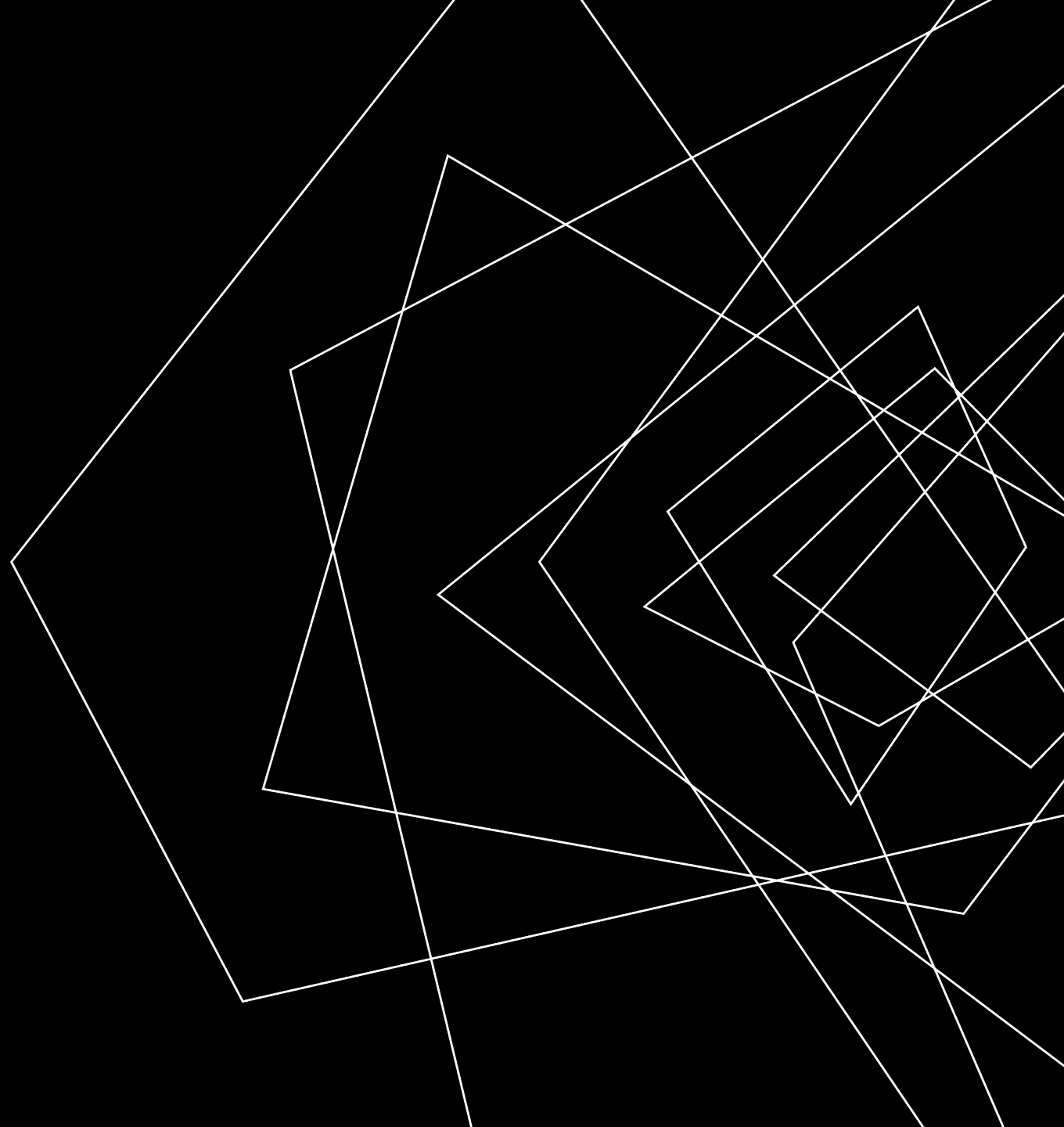
Md. Golam Saqlain – 2030404

Ikterab Yazda – 1910970

SM Tanzim Tuhin – 2022302

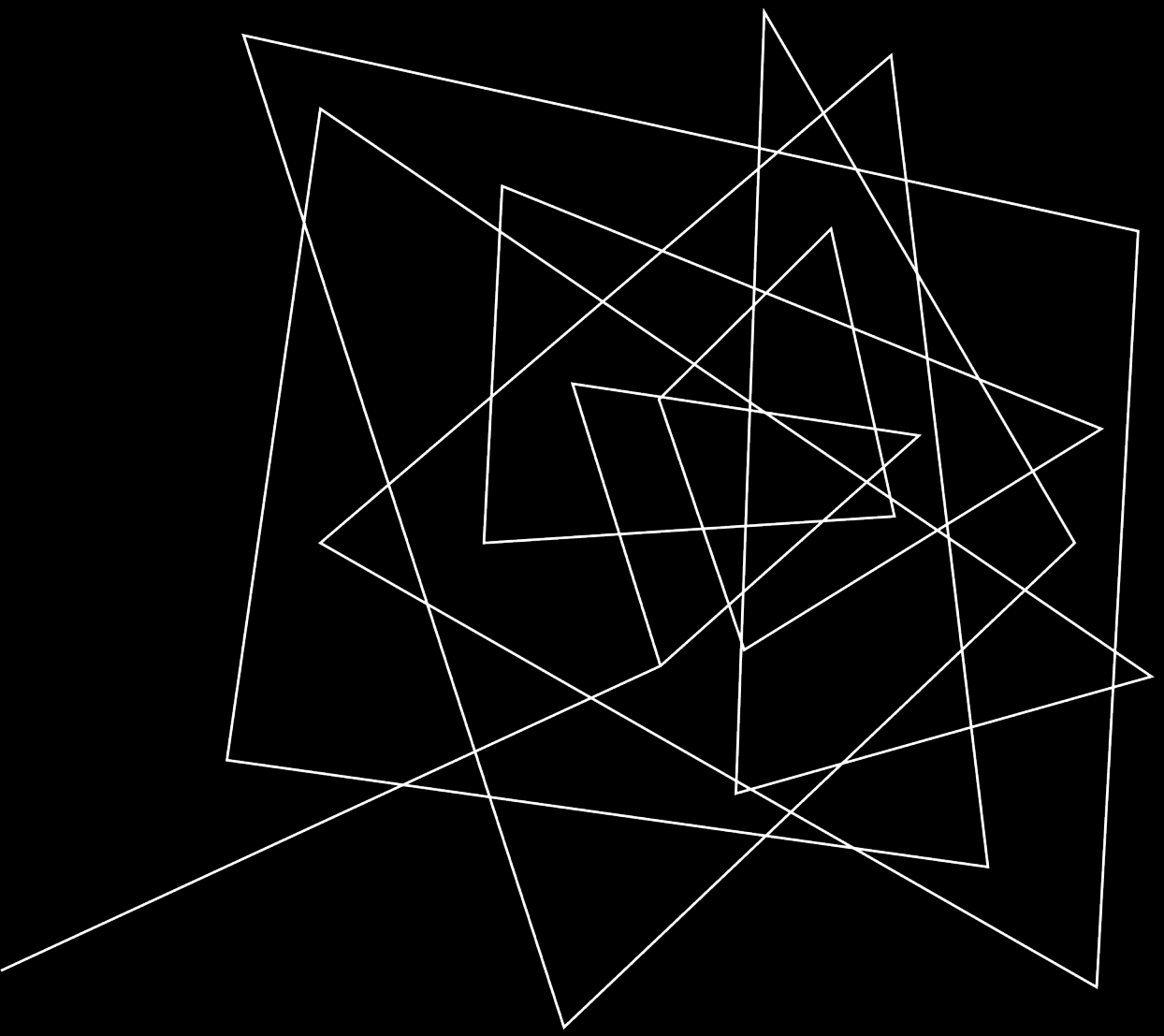
Monjurul Hoque Sharnav – 1821422

Ashir Abdal Ravee - 2020646



Introduction

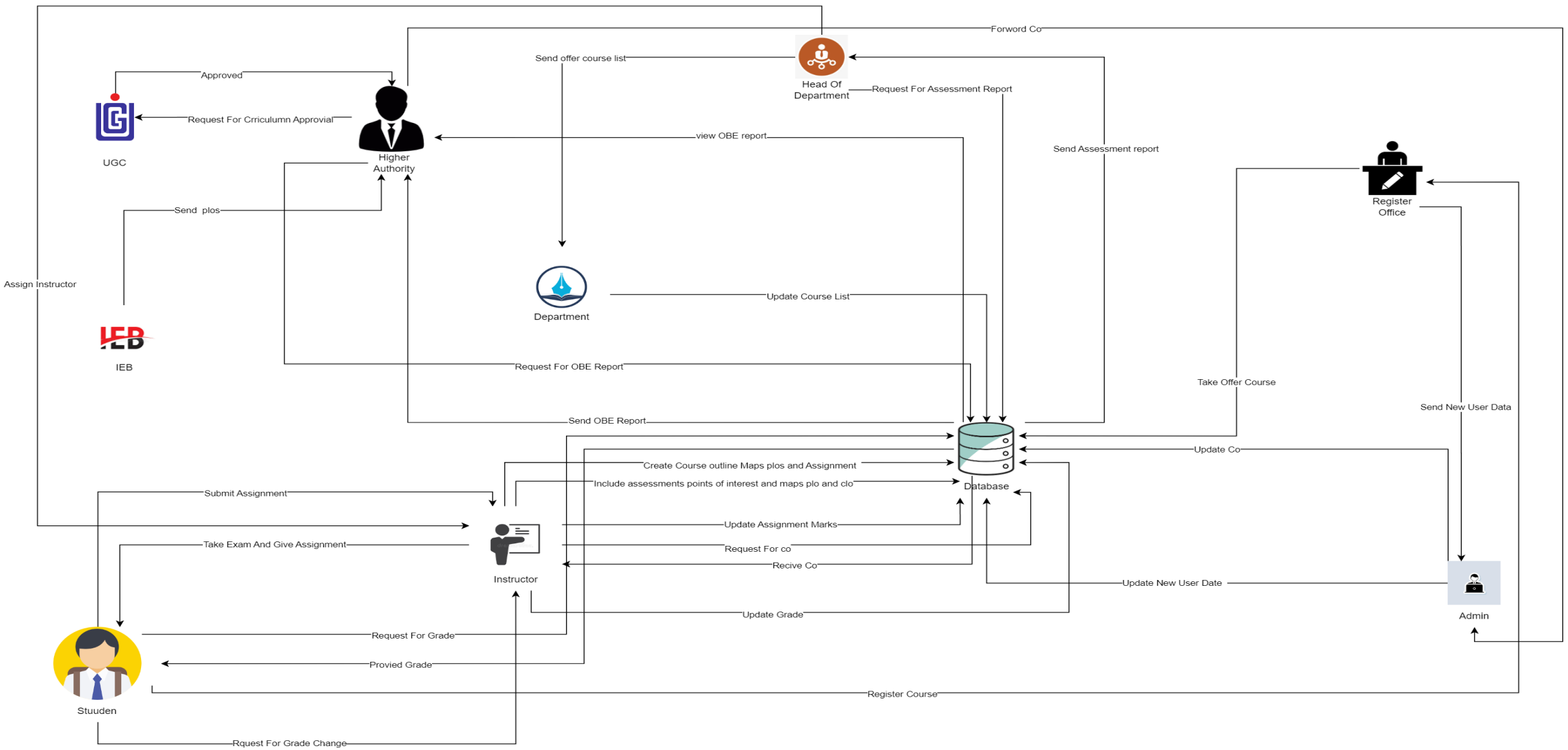
A database management system designed to help teachers, administrators, department heads, and deans manage and track the academic progress of students. With the ability to automate administrative processes, such as grade reporting and attendance tracking, this system enables early identification of academic difficulties and ensures each student receives the assistance needed to succeed. Our major goal is to provide a thorough platform for tracking and evaluating student performance and growth, allowing instructors to enter and record grades for various tasks, tests, and examinations, while also enabling students to monitor their progress over time. With SpmsV4, relevant documents and data are stored in the database to evaluate the performance of stakeholders, and students have access to a variety of analytical data.



SPMS

STUDENT PERFORMANCE MONITORING SYSTEM

Rich Picture (Existing)



Existing Process and Six Element Analysis

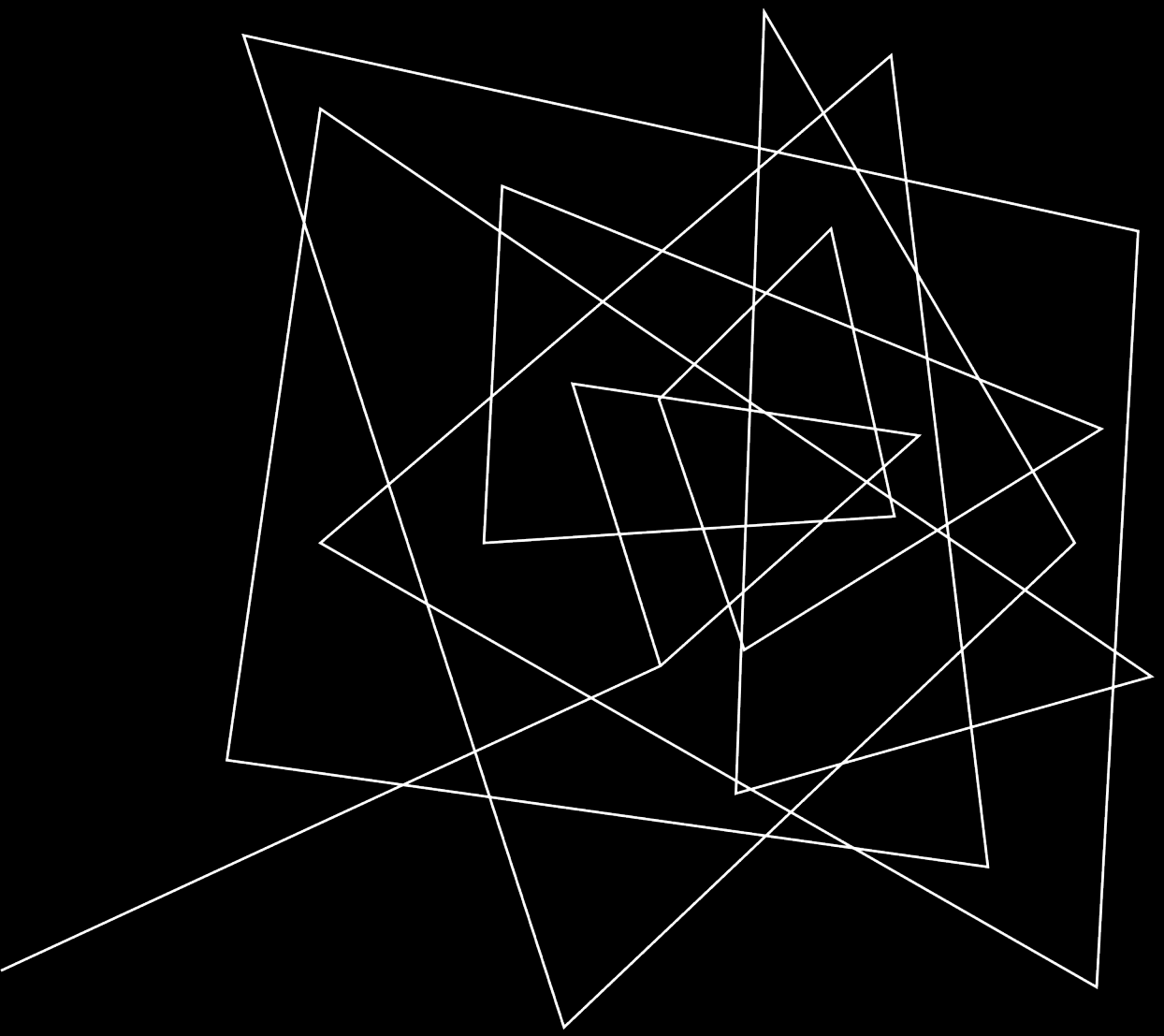
Six Element Analysis -> Existing System

From Rich picture we identified 7 key processes:

1. Calculate Course Outcomes (COs) and Program Learning Outcomes (PlOs),
2. Record Student Evaluation Data,
3. Generate OBE Marksheet and Course Evaluation Report,
4. View result and download Transcript,
5. Create Student/Faculty/Staff accounts and insert/adjust necessary data,
6. Analyze student’s records, OBE Marksheets and Course evaluation reports over time tosee student performance pattern and
7. Review and grade change request.

Problem Analysis

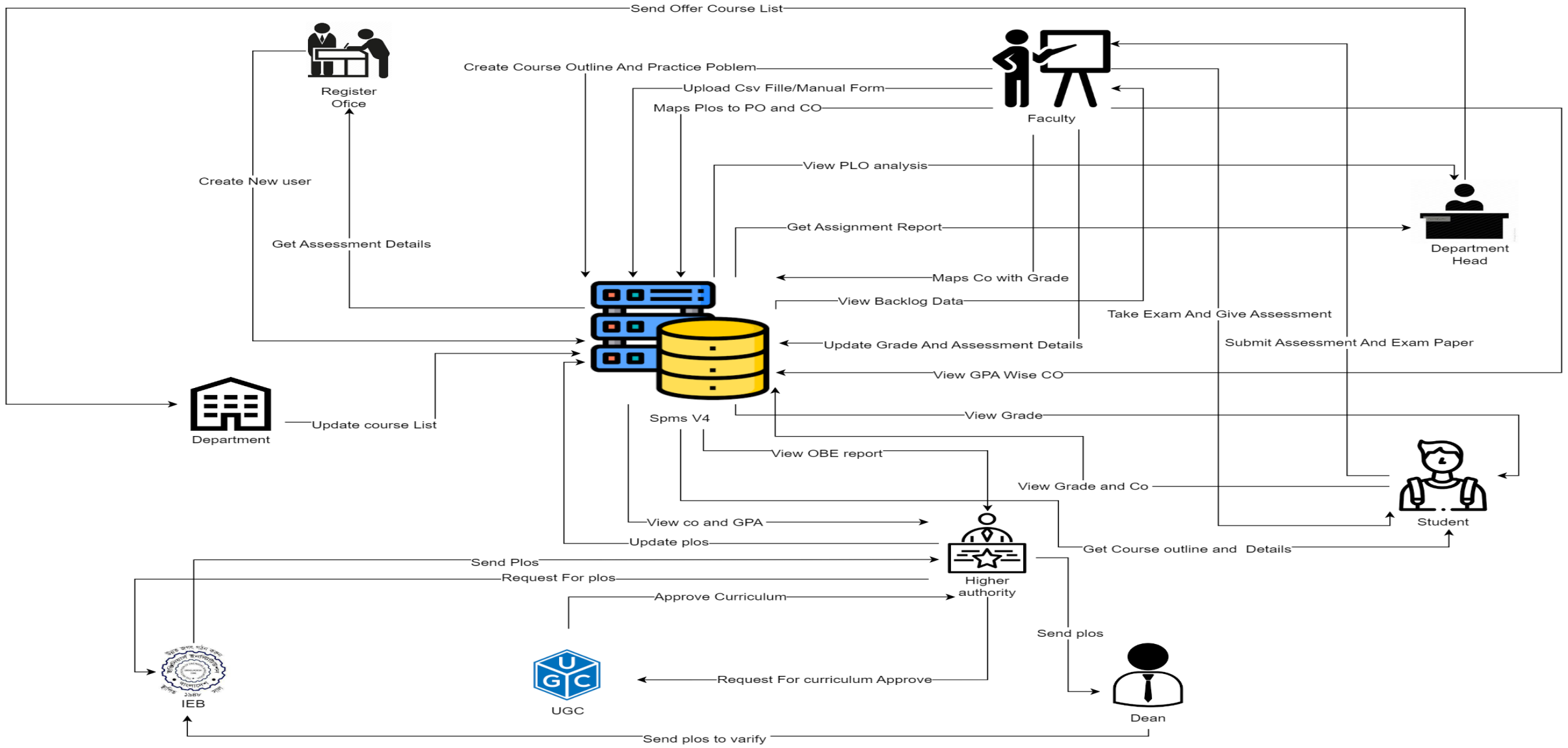
Process Name	Stakeholders	Concerns(Problems)	Analysis(Reason of Problems)	Proposed Solution
Record student evaluation data	1. Faculty Member 2. Student 3. Administration	1. Project work, assignments, question papers, and answer scripts condition 2. Giving and Receiving process 3. Storage Problem	1. The hard copies of answer scripts, project reports, and question papers that are being stored physically may get damaged or lost 2. The process of checking the answer script or receiving it and then recording it makes the process slow. 3. Physical storage maybe a problem for hard copies	The answer scripts and question paper are stored in the database, so it will be easy for him to find and search by student id. Which will be time reducing. Online submission of answer scripts saves time. Submitting online will be a good option for reducing storage problem
Generate OBE Marksheet and Course Evaluation Report	1. Faculty Member 2. Student 3. Administration	1. Storing hard copies and soft copies becomes hard to manage	1. Storing softcopies and hardcopies can become extremely difficult to manage when the organization has been operating for years. It also gets increasingly tedious to track documents to study student performance trends for a certain timeline. Updating information for a specific document would require tracking them, which in turn would make it harder to retrieve them and would require personnel to update various copies.	These problems can be fixed by maintaining these data tables in our database and giving necessary departments, offices, and outside parties (IEB) access via a user



SPMS V4

STUDENT PERFORMANCE MONITORING SYSTEM

Rich Picture (Proposed)



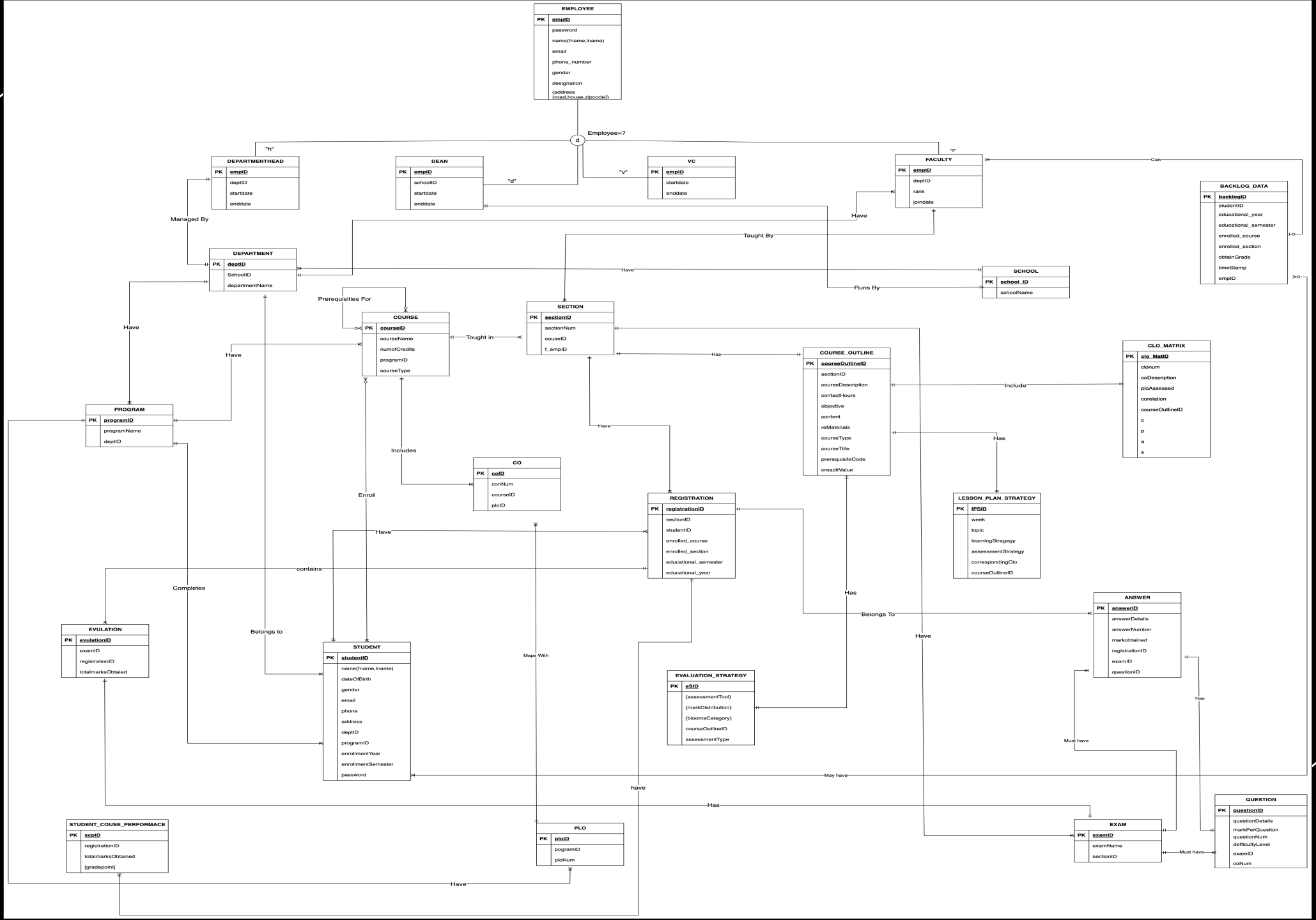
Proposed Process Six Element Analysis

Proposed process and Six Element Analysis

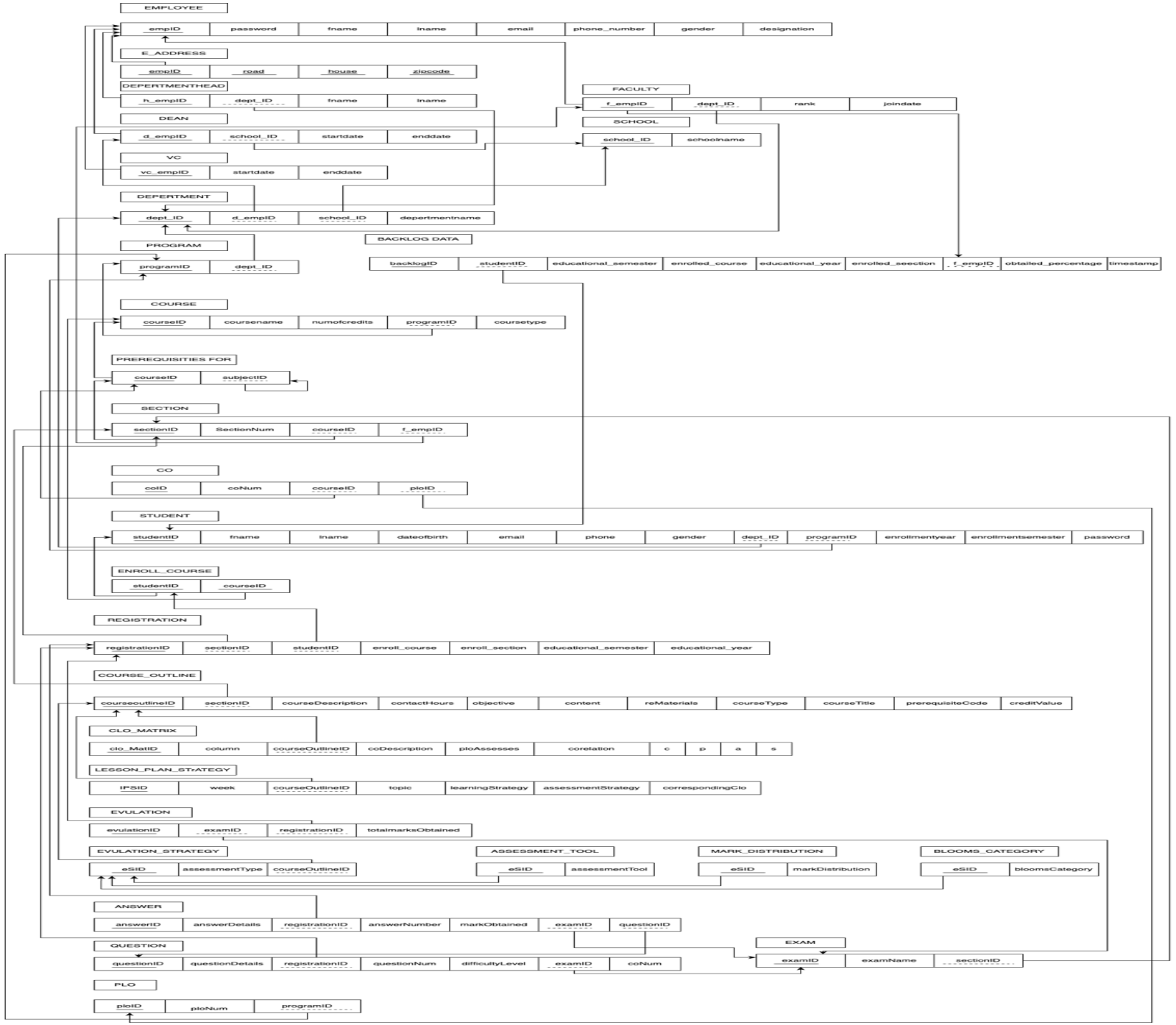
From Rich picture we identified 10 key processes:

1. Make and deliver course outlines and lesson plans,
2. Statistics of student enrollment under VC / Dean / Head of Department supervision,
3. Include question sheets of quizzes, assignments and exams and evaluate replays of answer sheet,
4. Statistics of student performance patterns by GPA within any course,
5. Statistics of student performance patterns by GPA under faculty supervision,
6. Statistics of student performance patterns by GPA under VC / Dean / Head of Department supervision,
7. Course Outcome (CO) and Program Outcome (PO),
8. Program Learning Outcome (PLO),
9. Statistics of Courses, Programs, Departments and Schools CLO and PLO and
10. Statistics of average CLO / PLO of students under Department Supervisions of total number of attempts.

Entity Relationship Diagram (ERD)



Relational Schema



Normalization

EMPLOYEE (e)	empID	e1
	password	e2
	name	e3
	email	e4
	phone_number	e5
	gender	e6
	designation	e7
	address	e8
VC (v)	vc_empID	v1
	startdate	v2
	enddate	v3
DEAN (n)	d_empID	n1
	schoolID	sc1
	startdate	n2
	enddate	n3
DEPARTMENTHEAD (h)	h_empID	h1
	dept_ID	d1
	startdate	h2
	enddate	h3
FACULTY (f)	f_empID	f1
	dept_ID	d1
	rank	f2
	joindate	f3
SCHOOL (sc)	school_ID	sc1
	schoolname	sc2
DEPARTMENT (d)	dept_ID	d1
	schoolID	sc1
	departmentname	d2
PROGRAM (p)	programID	p1
	programName	p2
	dept_ID	d1
COURSE (c)	courseID	c1
	courseName	c2
	numofCredits	c3
	programID	p1
	courseType	c4
SECTION (st)	sectionID	st1
	sectionNum	st2
	courseID	c1
	f_empID	f1
REGISTRATION (r)	registrationID	r1
	sectionID	st1
	studentID	s1
	enrolled_course	r2
	enrolled_section	r3
	educational_semester	r4
	educational_year	r5
STUDENT (s)	studentID	s1
	name	s2
	dateOfBirth	s3
	gender	s4
	email	s5
	phone	s6
	address	s7
	deptID	d1
	programID	p1
	enrollmentYear	s8
	enrollmentSemester	s9
	password	s10
COURSE_OUTLINE (cu)	courseOutlineID	cu1
	sectionID	st1
	courseDescription	cu2
	contactHours	cu3
	objective	cu4
	content	cu5
	reMaterials	cu6
	courseType	cu7
	courseTitle	cu8
	prerequisiteCode	cu9
	craditValue	cu10

LESSON_PLAN_STRATEGY (l)	lPSID	l1
	week	l2
	topic	l3
	learningStragegy	l4
	assessmentStragegy	l5
	corrospondingClo	l6
	courseOutlineID	cu1
EXAM (ex)	examID	ex1
	examName	ex2
	sectionID	st1
QUESTIION (q)	questionID	q1
	questionDetails	q2
	markPerQuestion	q3
	questionNum	q4
	deficultyLevel	q5
	examID	ex1
	coNum	q6
ANSWER (a)	answerID	a1
	answerDetails	a2
	answerNumber	a3
	markobtained	a4
	registrationID	r1
	questionID	q1
	examID	ex1

EVALUATION_STRATEGY (es)	eSID	es1
	assessmentTool	es2
	markDistribution	es3
	bloomsCategory	es4
	courseOutlineID	cu1
	assessmentType	es5
EVALUATION (ev)	evaluationID	ev1
	examID	ex1
	registrationID	r1
	totalmarksObtained	ev2
CLO_MATRIX (cm)	clo_MatID	cm1
	clonum	cm2
	coDescription	cm3
	ploAssessed	cm4
	corelation	cm5
	courseOutlineID	cu1
	c	cm6
	p	cm7
	a	cm8
	s	cm9
CO (co)	coID	co1
	coNum	co2
	courseID	c1
	ploID	pl1

PLO (pl)	ploID	pl1
	programID	p1
	ploNum	pl2
STUDENT_COURSE_PERFORMANCE (sp)	scpID	sp1
	registrationID	r1
	totalmarksObtained	sp2
BACKLOG_DATA (b)	backlogID	b1
	studentID	s1
	enrolled_course	r2
	enrolled_section	r3
	educational_semester	r4
	educational_year	r5
	timestamp	b2
	f_empID	f1

Functional Dependences

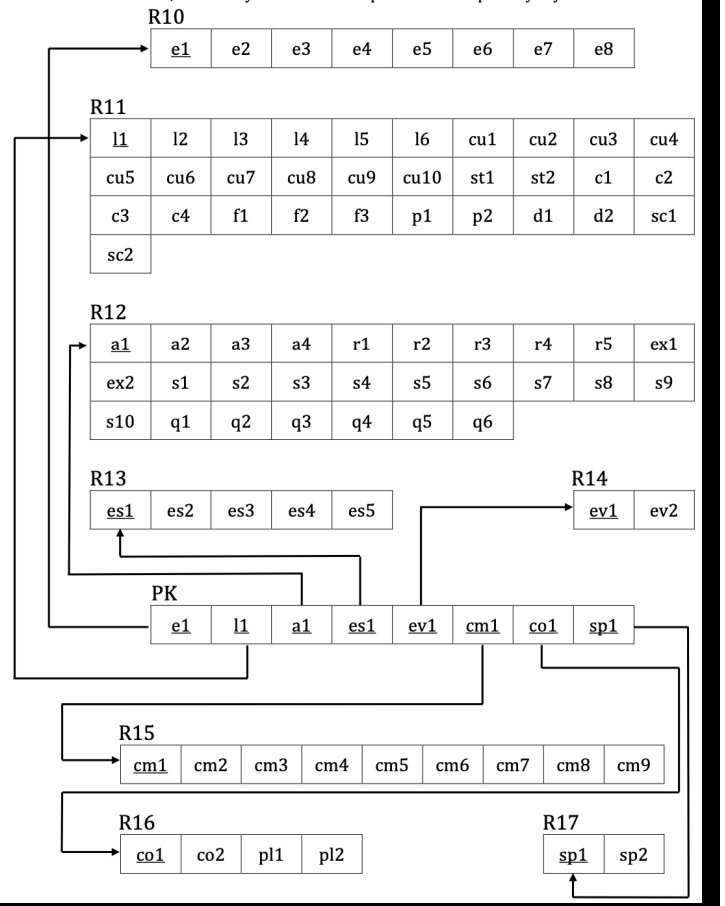
e1	e2, e3, e4, e5, e6, e7, e8
v1	v2, v3
n1	n1, n2, n3, sc1
h1	h2, h3, d1
f1	f2, f3, d1
sc1	sc2
d1	d2, sc1
p1	p2, d1
c1	c2, c3, c4, p1
st1	st2, st3, c1, f1
r1	r2, r3, r4, r5, st1, s1
s1	s2, s3, s4, s5, s6, s7, s8, s9, s10, d1, p1
cu1	cu2, cu3, cy4, cu5, cu6, cu7, cu8, cu9, cu10, st1
l1	l2, l3, l4, l5, l6, cu1
ex	ex2, st1
q1	q2, q3, q4, q5, q6, ex1
a1	a2, a3, a4, r1, q1, ex1
es1	es2, es3, es4, es5, cu1
ev1	ev2, ex1, r1
cm1	cm2, cm3, cm4, cm5, cm6, cm7, cm8, cm9, cu1
co1	co2, c1, p1
pl1	pl2, p1
sp1	sp2, r1
b1	b2, s1, f1, r2, r3, r4, r5

1NF:

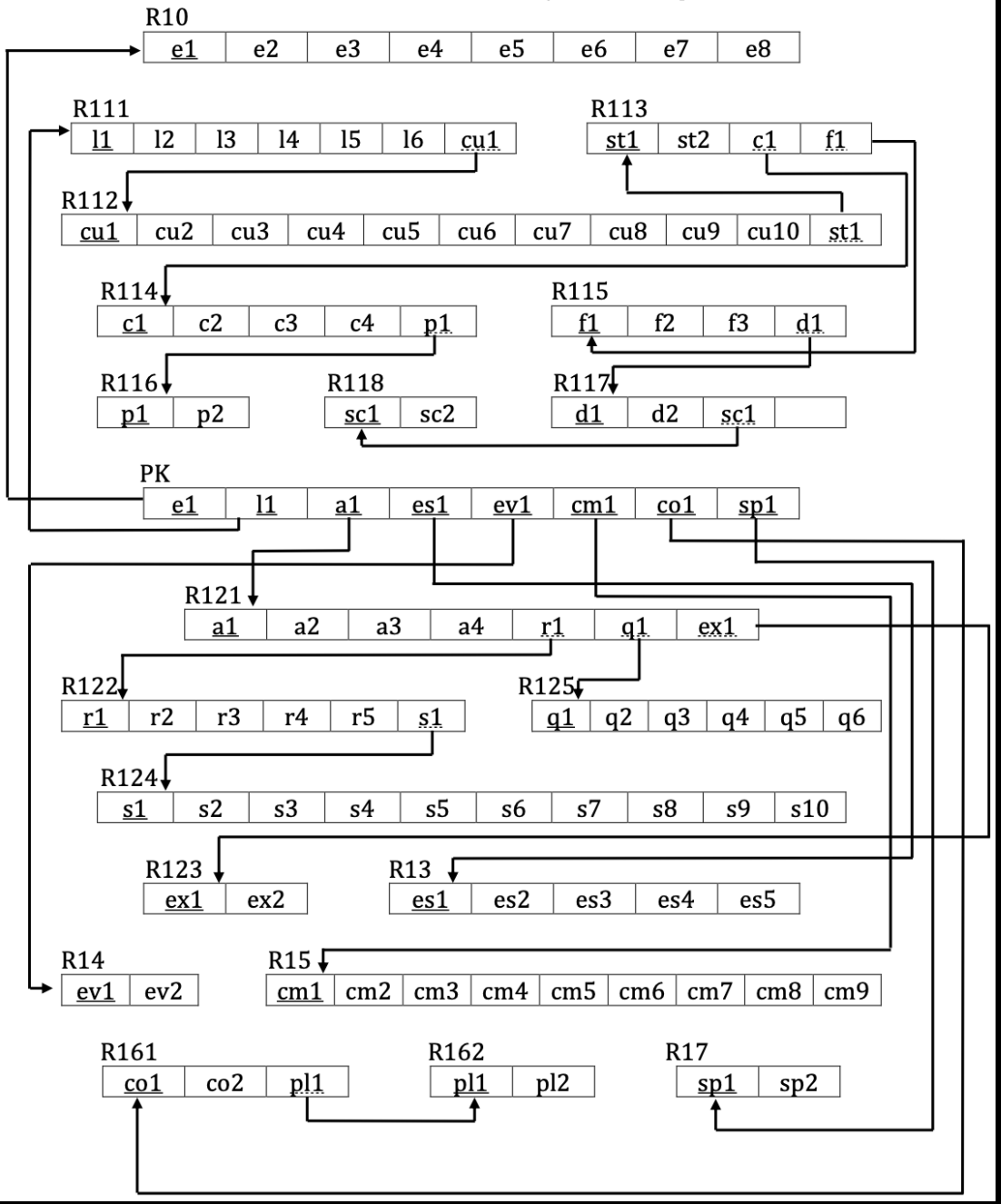
R1 =

e1	l1	a1	es1	ev1	cm1	co1	sp1	e2	e3
e4	e5	e6	e7	e8	l2	l3	l4	l5	l6
cu1	cu2	cu3	cu4	cu5	cu6	cu7	cu8	cu9	cu10
st1	st2	c1	c2	c3	c4	f1	f2	f3	p1
p2	d1	d2	sc1	sc2	a2	a3	a4	r1	r2
r3	r4	r5	q1	q2	q3	q4	q5	q6	ex1
ex2	s1	s2	s3	s4	s5	s6	s7	s8	s9
s10	es2	es3	es4	es5	ev2	cm2	cm3	cm4	cm5
cm6	cm7	cm8	cm9	co2	pl1	pl2	sp2		

2NF: Here, all non-key attributes are dependent on the primary key.



3NF: Here, this relation is in 2NF and don't have any transitive dependencies.



Data Dictionary

Employee_T

Name	Data Type	Size	Remark
nemployeeID	INTEGER	11	This is the primary key for Employee table. E.g.: "1801"
cpassword	VARCHAR	10	This is the password of the employee
cfirstname	VARCHAR	50	This is the first name of the faculty. E.g.: "Noor"
clastname	VARCHAR	50	This is the last name of the faculty. E.g.: "Sadman"
cemail	VARCHAR	30	This is the email of the employee. E.g.: "arnoyk123sets@iub.edu.bd."
cphone_number	VARCHAR	11	This is the phone number of the employee. E.g.: "01XXXXXXXXX".
cgender	VARCHAR	6	This is the gender of the employee. E.g.: "Male".
cdesignation	VARCHAR	20	Employee hold an office or post. E.g.: "an employee can be a department head, dean, vc, faculty".
caddress	VARCHAR	50	This is the address of the employee. E.g: "House 12, Road 1, Block F, Bashundhara RA

VC_T

Name	Data Type	Size	Remark
nv_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g.: "4250"
dstartDate	DATE	DD MM YYYY	This is starting date for the VC. E.g.: "01-06-2020"
dendDate	DATE	DD MM YYYY	This is the date VC retire from his post. E.g.: "01-03-2023

SCHOOL_T

Name	Data Type	Size	Remark
cschoolID	VARCHAR	5	This is the primary key of School. E.g: "SETS"
cschoolName	VARCHAR	50	This is the name of the school. E.g.: "School of Engineering, Technology & Science".

Data Dictionary cont.

DEPARTMENT_T

Name	Data Type	Size	Remark
cdepartmentID	VARCHAR	3	This is the primary key for the Department table. E.g: "CSE"
cdepartmentName	VARCHAR	50	This is the name of the department. E.g: "Computer Science and Engineering".
cSchoolID	VARCHAR	5	This is a foreign key from the school table. E.g: "SETS".

DEPARTMENTHEAD_T

Name	Data Type	Size	Remark
nh_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g.: "4228"
cdepartmentID	VARCHAR	3	This is the departmentID of the department HEAD manages. E.g.: "CSE"
dstartdate	DATE	DD MM YYYY	This is starting date. E.g.: "01-07-2021"
dendtdate	DATE	DD MM YYYY	This is the date HEAD retire from his post. E.g.: "01-03-2024"

DEAN_T

Name	Data Type	Size	Remark
nd_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g: "4250"
cschoolID	VARCHAR	5	This is the SchoolID of the school DEAN manages. E.g: "SETS"
dstartdate	DATE	DD MM YYYY	This is starting date. E.g: "11-03-2020"
denddate	DATE	DD MM YYYY	This is the date DEAN retire from his post. E.g: "12-03-2024"

Data Dictionary cont.

FACULTY_T

Name	Data Type	Size	Remark
nf_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g.: "4450"
cdepertmentID	VARCHAR	3	This is the DepartmentID of the department faculty belongs to. E.g.: "CSE"
crank	VARCHAR	30	This is the rank of the faculty. E.g: "Assistant Professor"
djoindate	DATE	DD MM YYYY	This is starting date. E.g: "01-03-2022"

PROGRAM_T

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

COURSE_T

Name	Data Type	Size	Remark
ccourseID	VARCHAR	6	This is the Primary Key for the Course. E.g: "CSE203"
ccourseName	VARCHAR	40	This is the name of the Course. E.g: "Database Management"
nnumofCredits	INTEGER	11	This is the number of credits for the Course. E.g: "3"
ccourseType	VARCHAR	10	This is the type of the Course. E.g: "Core"
nprogramID	INTEGER	11	This is the foreign key from the program table. E.g: "1"

Data Dictionary cont.

PRE_REQ_COURSE_T

Name	Data Type	Size	Remark
npreReqID	INTEGER	11	This is the primary key for this table. E.g.: "1"
cpreReqCourseID	VARCHAR	6	This is the id of prereqcourse. E.g:" CSE101"
ccourseID	VARCHAR	6	This is the foreign key from the course table. E.g: "CSE203"

SECTION_T

Name	Data Type	Size	Remark
nsectionID	INTEGER	11	This is the Primary Key for Section. E.g: "1"
nsectionNum	INTEGER	11	This is the section number. E.g: "1"
nf_empID	INTEGER	11	This is the foreign key from Faculty table. E.g: "1801"
ccourseID	VARCHAR	6	This is the foreign key from the Course table. E.g: "CSE101"

COURSE_OUTLINE_T

Name	Data Type	Size	Remark
ncourseOutlineID	INTEGER	11	This is the primary key for this table
nSectionID	INTEGER	11	This is the foreign key from the section table
cSemester	VARCHAR	30	This is the semester name. E.g.: spring 2021
nsectionNum	INTEGER	11	
mtcourseDescription	MEDIUMTEXT		This is the description of the course
mtobjective	MEDIUMTEXT		This is the objective of the course
mtcontent	MEDIUMTEXT		This is the content of the course
mtrefMaterials	MEDIUMTEXT		This is the reference material
ccourseTitle	VARCHAR	1000	This is the title of the course
cprerequsiteCode	VARCHAR	6	This is the prerequisite course code
ncreditValue	INTEGER	11	This is the credit value of the course

Data Dictionary cont.

CLO_MATRIX_T

Name	Data Type	Size	Remark
nclo_MatID	INTEGER	11	This is the primary key for this table
nclonum	INTEGER	11	This is the clo number
mtcoDescription	MEDIUMTEXT		This is the co-description
cploAssessed	VARCHAR	10	This is the name of the plo assessed
ncorrelation	INTEGER	11	This is the correlation value or number
ncourseOutlineID	INTEGER	11	This is the foreign key from the course outline table
nc	INTEGER	11	This is the bloom's category level
np	INTEGER	11	This is the bloom's category level
na	INTEGER	11	This is the bloom's category level
ns	INTEGER	11	This is the bloom's category level

CO_T

Name	Data Type	Size	Remark
ncoID	INTEGER	11	This is the primary key for the CO table. E.g: "CO1".
ncoNum	INTEGER	11	This is the CO number. E.g: 1,2 etc.
ccourseID	VARCHAR	6	This is the foreign key from the Course table. E.g: "CSE303"
cploID	VARCHAR	5	This is the foreign key from the PLO table. E.g: "PLO1"

Data Dictionary cont.

STUDENT_T

Name	Data Type	Size	Remark
nstudentID	INTEGER	11	This is the primary key for the Student table. E.g: "2022315".
cfirstname	VARCHAR	30	This is the first name of the student. E.g: "Imran".
clastname	VARCHAR	30	This is the last name of the student. E.g: "Hasan".
ddateOfBirth	DATE	DD MM YYYY	This is the birth date of the student. E.g: "21-12-2001".
cgender	VARCHAR	6	This is the gender of the student. E.g: "Male".
cemail	VARCHAR	30	This is the email of the student. E.g: "2022315@iub.edu.bd"
cphone	VARCHAR	11	This is the phone of the student. E.g: "01XXXXXXXXX".
caddress	VARCHAR	50	This is the address of the student. E.g: "House 12, Road 1, Block F, Bashundhara RA
cdepertmentID	VARCHAR	3	This is the foreign key from the Department table. E.g: "CSE"
nprogramID	INTEGER	11	This is the foreign key from the Program table. E.g: "1"
cenrollmentSemester	VARCHAR	6	This is the enrollment semester of the student. E.g: "Summer"
yenrollmentYear	YEAR	YYYY	This is enrollment year of the student. E.g: "2020"
cpassword	VARCHAR	10	This is the password of the student.

REGISTRATION_T

Name	Data Type	Size	Remark
nregistrationID	INTEGER	11	This is the Primary Key for Registration. E.g: "0101010101"
nsectionID	INTEGER	11	This is the foreign key from section table
nstudentID	INTEGER	11	This is the foreign key from student table
nenrolled_course	INTEGER	11	This is the number of course a student can enroll
nenrolled_section	INTEGER	11	This is the number of sections a student can enroll.
yeducationa_year	YEAR	YYYY	This is the educational year. E.g: 2022
cEducational_semester	VARCHAR	6	This is the educational semester of the student. E.g: "Summer"

EXAM_T

Name	Data Type	Size	Remark
nexamID	INTEGER	11	This is the primary key for this table
cexamName	VARCHAR	30	This is the name of the exam
nsectionID	INTEGER	11	This is the foreign key from section table

Data Dictionary cont.

EVALUATION_T

Name	Data Type	Size	Remark
nevaluationID	INTEGER	11	This is the primary key for this table
nregitrationID	INTEGER	11	This is the foreign key from registration table
ntotalmarks	INTEGER	11	This is the total marks achieved by the student in a specific exam
nexamID	INTEGER	11	This is the foreign key from exam table

EVALUATION_STRATEGY_T

Name	Data Type	Size	Remark
neSID	INTEGER	11	This is the primary key for this table
ncourseOutlineID	INTEGER	11	This is the foreign key from course outline table

ASSESSMENT_TOOL_T

Name	Data Type	Size	Remark
nasID	INTEGER	11	This is the primary key for this table
neSID	INTEGER	11	This is the foreign key from evaluation strategy table

Data Dictionary cont.

MARK_DISTRIBUTION_T

Name	Data Type	Size	Remark
nmdID	INTEGER	11	This is the primary key for this table
neSID	INTEGER	11	This is the foreign key from evaluation strategy table

BLOOMS_CATEGORY_T

Name	Data Type	Size	Remark
nbcID	INTEGER	11	This is the primary key for this table
neSID	INTEGER	11	This is the foreign key from evaluation strategy table

LESSON_PLAN_STRATEGY_T

Name	Data Type	Size	Remark
nlpsID	INTEGER	11	This is the primary key of the table
nweek	INTEGER	11	This is the week number
mttopic	MEDIUMTEXT		This is the topic name
mtlearningStrategy	MEDIUMTEXT		This is the lesson plan strategy of that topic
mtcassessmentStrategy	MEDIUMTEXT		This is the assessment strategy of that topic
ncourseOutlineID	INTEGER	11	This is the foreign key from course outline table
ccorrespondingClo	VARCHAR	10	This is the corresponding clo For the corresponding course.

Data Dictionary cont.

PLO_T

Name	Data Type	Size	Remark
nploID	INTEGER	11	This is the primary key for Program Learning Outcome. E.g: "PLO2"
nploNum	INTEGER	11	This is the PLO number. E.g: "2"
nprogramID	INTEGER	11	This is a foreign key from Program table. E.g: "2"

STUDENT_COURSE_PERFORMANCE_T

Name	Data Type	Size	Remark
nscpID	INTEGER	11	This is the primary key for this table
nregistrationID	INTEGER	11	This is the foreign key from registration table
ntotalmarksObtained	INTEGER	11	This is the total marks obtained by the student
fgradePoint	FLOAT		This is the grade point achieved by the student

QUESTION_T

Name	Data Type	Size	Remark
nquestionID	INTEGER	11	This is the primary key of this table
mtquestionDetails	MEDIUMTEXT		This is the question
nmarkPerQuestion	INTEGER	11	This is the mark each question contains
nquestionNum	INTEGER	11	This is the number of the question
ndifficultyLevel	INTEGER	11	This is the difficulty level of the question
nexamID	INTEGER	11	This is the foreign key from exam table
ncoNum	INTEGER	11	This is the CO number of the question
ccourseID	VARCHAR	6	This is the foreign Key for this table. E.g: "CSE203"

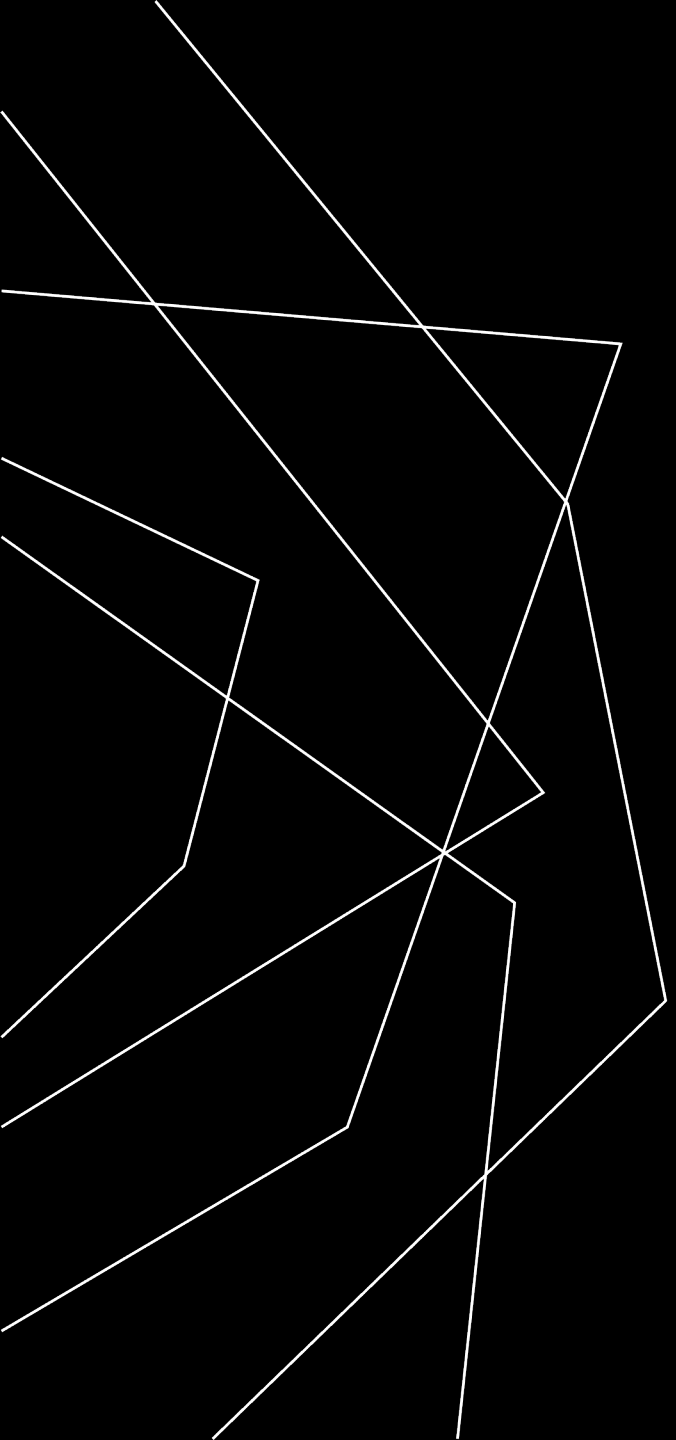
Data Dictionary cont.

ANSWER_T

Name	Data Type	Size	Remark
nanswerID	INTEGER	11	This is the primary key for this table
mtanswerDetails	MEDIUMTEXT		This is the answer details
nanswerNumber	INTEGER	11	This is the number of the answer
nmarkObtained	INTEGER	11	This is the mark obtained by the student for each answer
nregistrationID	INTEGER	11	This is the foreign key from registration table
nexamID	INTEGER	11	This is the foreign key from the exam table
nquestionID	INTEGER	11	This is the foreign Key from question table

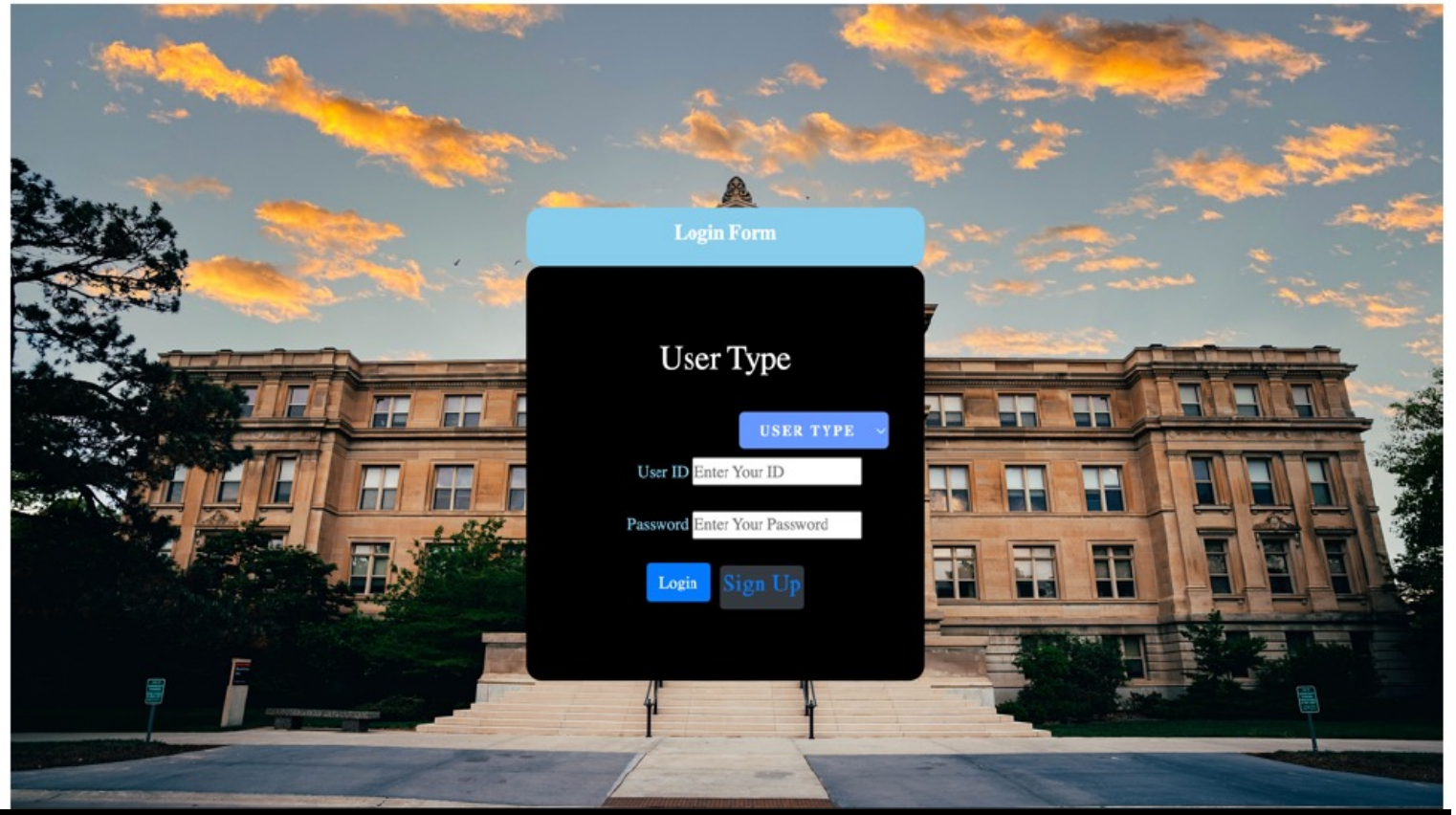
BACKLOG_DATA_T

Name	Data Type	Size	Remark
nbacklogID	INTEGER	11	This is the Primary Key for this table. E.g: “11108297”
nstudentID	INTEGER	11	This is the ID of student. E.g: “2022315”.
yeducationa_year	YEAR	YYYY	This is the educational year. E.g: 2022.
ceducational_semester	VARCHAR	6	This is the educational semester of the student. E.g: “Summer”
nenrolled_course	INTEGER	11	This is the number of course a student can enroll
nenrolled_section	INTEGER	11	This is the number of sections a student can enroll.
fobtainGrade	FLOAT		This is the obtained Percentage for a student
dtimestamp	DATE	DD MM YYYY	This is the timestamp for backlog data. E.g.:”21-12-2022”
nempID	INTEGER	11	This is the foreign key from the Employee table. E.g.: “4450”



Physical System Design

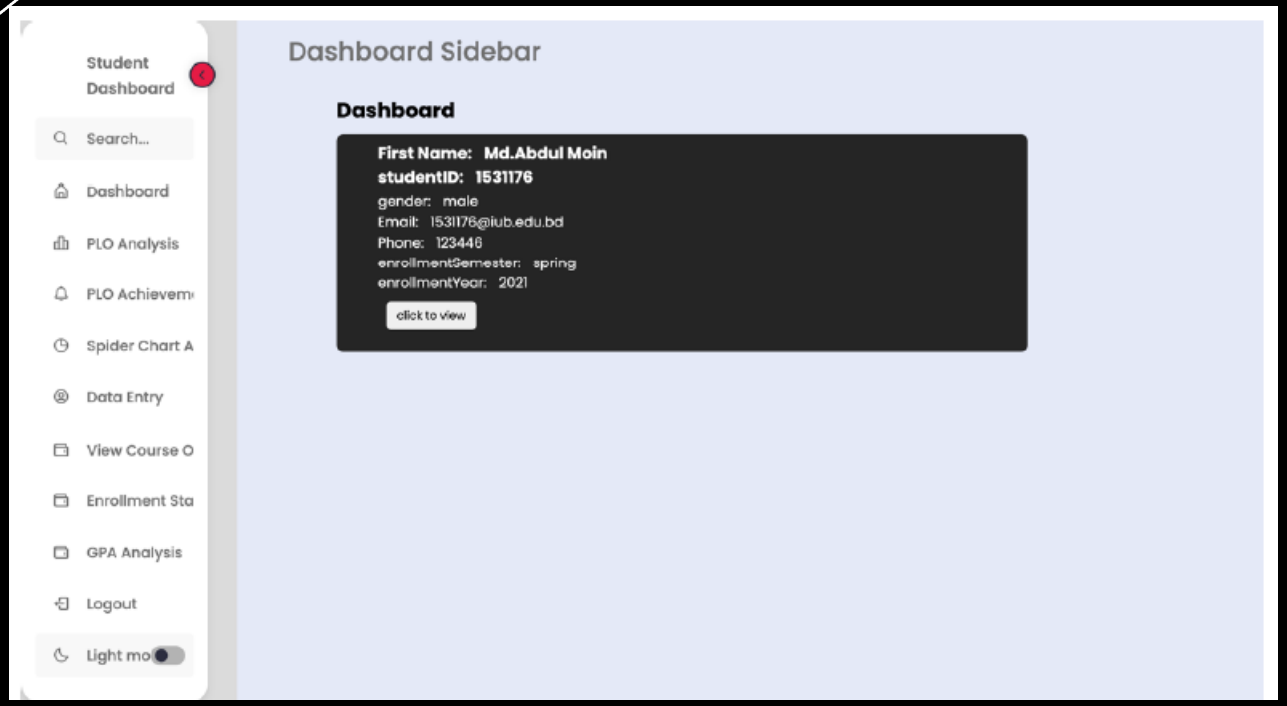
Input Forms



login.php

```php
<?php
 require_once(__DIR__."/testfunction.php");
 require_once(__DIR__."/connect.php");
session_start();
$invalid=0;
if($_SERVER['REQUEST_METHOD']=='POST'){


    echo pre();

    $userType=$_POST['userType'];
    $ID=$_POST['ID'];
    $password=$_POST['password'];

    if($userType=='faculty'){
        $sql="SELECT * from employee_t where employeeID='$ID' and password='$password'";
        var_dump($_POST);
        $result=mysqli_query($con,$sql);
        if($result){
            $num=mysqli_num_rows($result);
            if($num>0){
                $invalid=0;
                $_SESSION['ID']=$ID;
                $_SESSION['userType']='faculty';
                header('location:employee_dashboard.php');
            }
        }
    }elseif($userType=='student'){
      var_dump($_POST);
        $sql="SELECT * from student_t where studentID='$ID' and password='$password'";
        $result=mysqli_query($con,$sql);
        if($result){
            $num=mysqli_num_rows($result);
            if($num>0){
                $invalid=0;
                $_SESSION['ID']=$ID;
                $_SESSION['userType']='student';
                header('location:estudent_dashboard.php');
            }
        }
    }else{
        $invalid=1;
    }
}
?>
```

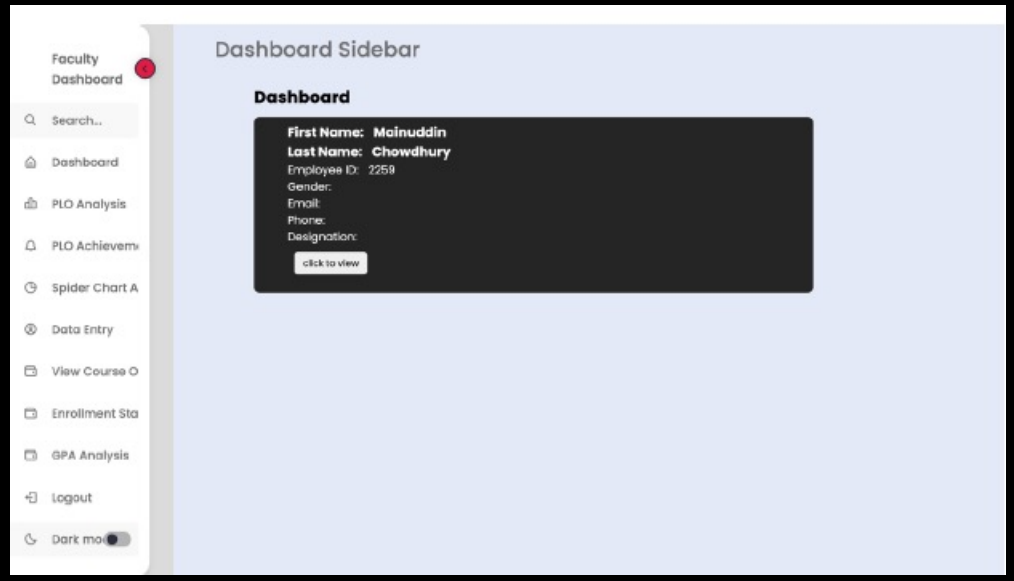
Input Forms cont.



estudent_dashboard.php

```php

<?php
require_once(__DIR__."/testfunction.php");
require_once(__DIR__."/connect.php");
require_once(__DIR__."/user_header.php");
// $_SESSION['dashboard_referrer'] = 'estudent_dashboard.php';

// if (!empty($_SERVER['HTTP_REFERER'])) {
//     $_SESSION['dashboard_referrer'] = $_SERVER['HTTP_REFERER'];
// }

//echo pre($_SESSION);
if (isset($_POST['submit'])) {
    $studentID = $_POST['studentID'];
} elseif (isset($_SESSION['ID'])) {
    $studentID = $_SESSION['ID'];
}

// Fetch employee data
$student_data = null;
if (isset($studentID)) {
    $query = "SELECT * FROM student_t WHERE studentID = ?";
    $stmt = $con->prepare($query);
    $stmt->bind_param("s", $studentID);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows > 0) {
        $student_data = $result->fetch_assoc();
    }
}


?>
```

Input Forms cont.



employee_dashboard.php

```php

<?php
require_once(__DIR__."/testfunction.php");
require_once(__DIR__."/connect.php");
require_once(__DIR__."/user_header.php");
//echo pre($_SESSION);

if (isset($_POST['submit'])) {
    $employeeID = $_POST['employeeID'];
} elseif (isset($_SESSION['ID'])) {
    $employeeID = $_SESSION['ID'];
}

// Fetch employee data
$employee_data = null;
if (isset($employeeID)) {
    $query = "SELECT * FROM employee_t WHERE employeeID = $ID?";
    $stmt = $con->prepare($query);
    $stmt->bind_param("s", $employeeID);
    $stmt->execute();
    $result = $stmt->get_result();
    if ($result->num_rows > 0) {
        $employee_data = $result->fetch_assoc();
    }
}


?>
```

connect.php

```php
<?php

$HOSTNAME='localhost';
$USERNAME='root';
$PASSWORD='';
$DATABASE='spms';

$con=mysqli_connect($HOSTNAME,$USERNAME,$PASSWORD,$DATABASE);

if(!$con){
    die(mysqli_error($con));
}

?>
```

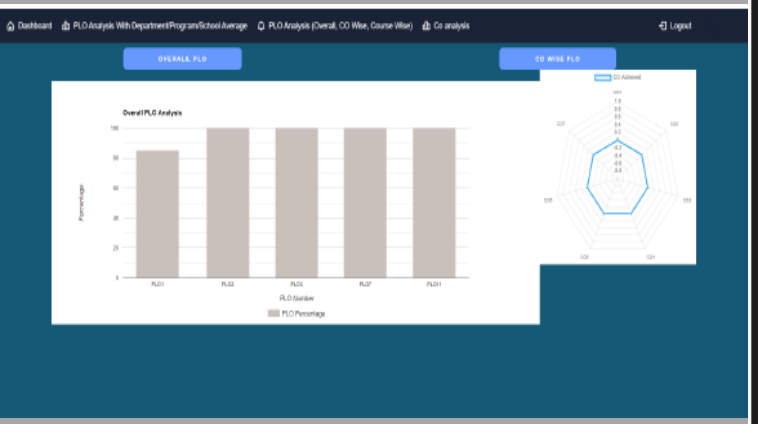
logout.php

```php
<?php
session_start();
session_unset();
session_destroy();
header('location:login.php');
?>

```

Output Forms


```php
<?php
require_once(__DIR__."/testfunction.php");
require_once(__DIR__."/connect.php");
require_once(__DIR__."/user_header.php");

if (!empty($_SERVER['HTTP_REFERER'])) {
    $_SESSION['dashboard_referrer'] = $_SERVER['HTTP_REFERER'];
}

// Set the dashboard link based on the previous page URL
if (isset($_SESSION['dashboard_referrer']) && strpos($_SESSION['dashboard_referrer'], 'estudent_dashboard.php') !== false) {
  $dashboardLink = 'estudent_dashboard.php';
} else {
  $dashboardLink = 'employee_dashboard.php';
}

echo "dashboardLink = " . $dashboardLink;

// current page
$dashboardLink = isset($_SESSION['dashboard_referrer']) && strpos($_SESSION['dashboard_referrer'], 'estudent_dashboard.php') !== false ?
'estudent_dashboard.php' : 'employee_dashboard.php';
echo '<a href="' . $dashboardLink . '">';
// Set the dashboard link based on the previous page URL
// if (isset($_SESSION['dashboard_referrer']) && strpos($_SESSION['dashboard_referrer'], 'estudent_dashboard.php') !== false) {
//   $dashboardLink = 'estudent_dashboard.php';
// } else {
//   $dashboardLink = 'employee_dashboard.php';
// }

//echo pre($_SESSION);
if (isset($_POST['submit'])) {
    $studentID = $_POST['studentID'];
} elseif (isset($_SESSION['ID'])) {
    $studentID = $_SESSION['ID'];
}

?>
```

Output Forms cont.



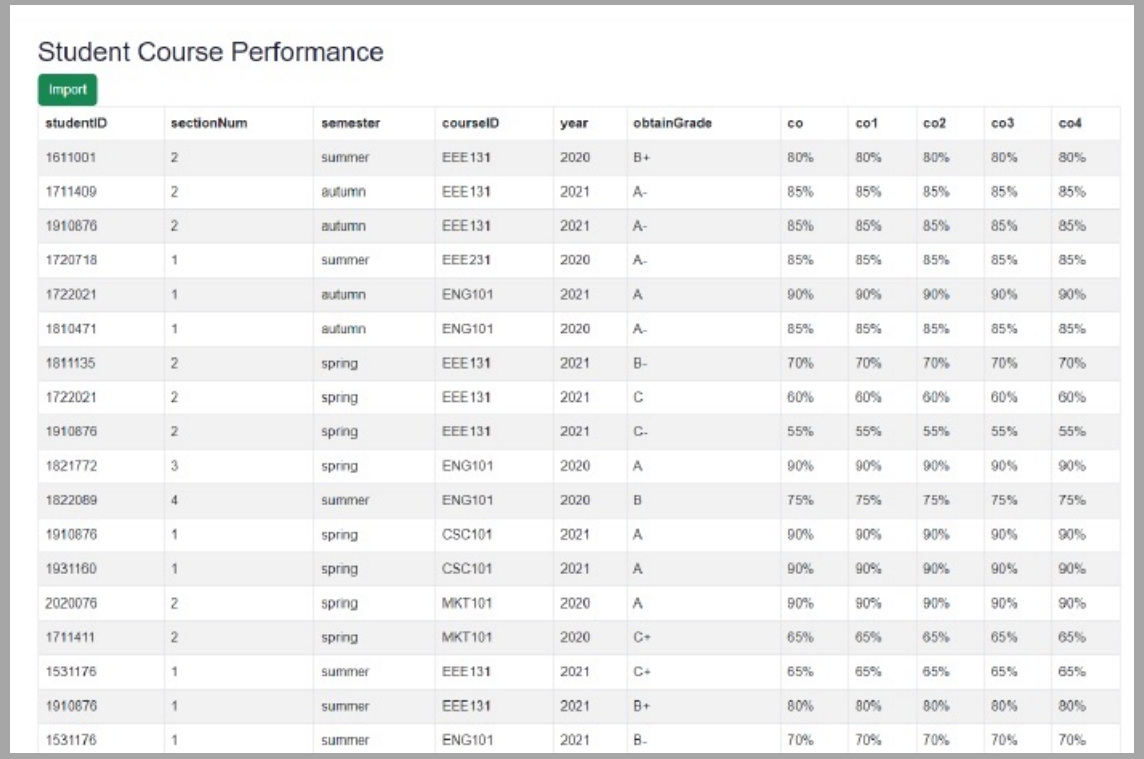
spiderChart.php

```php
<?php
if (isset($_POST['submit'])) {
  $studentID = $_POST['studentID'];
} elseif (isset($_SESSION['ID'])) {
  $studentID = $_SESSION['ID'];
}
?>


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

    $result = mysqli_query($con, $sql);

    $po = array();
    $percent = array();

    while ($data = mysqli_fetch_array($result)) {

      array_push($po, "PO " . $data['poNum']);
      array_push($percent, $data['percent']);
    }

    ?>


    var po = <?php echo json_encode($po); ?>;
    var percent = <?php echo json_encode($percent); ?>;

    for (var i = 0; i < percent.length; i++) {
      percent[i] = parseFloat(percent[i]);
    }

    document.getElementById("chart-container").innerHTML='';
    document.getElementById("chart-container").innerHTML='<canvas style="background-color:white;height:500px;width:400px;" id="myChart"></canvas>';

    const ctx = document.getElementById('myChart');

    new Chart(ctx, {
      type: 'radar',
      data: {
        labels: po,
        datasets: [{
          label: 'PO Achieved',
          data: percent,
          fill: true,
          backgroundColor: 'rgba(54, 162, 235, 0.2)',
          borderColor: 'rgb(54, 162, 235)',
          pointBackgroundColor: 'rgb(54, 162, 235)',
          pointBorderColor: '#fff',
          pointHoverBackgroundColor: '#fff',
          pointHoverBorderColor: 'rgb(54, 162, 235)'
        }]
      },
      options: {
        elements: {
          line: {
            borderWidth: 3
          }
        }
      }
    });

  }
```

spiderChart.php

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

    $result = mysqli_query($con, $sql);

    $co = array();
    $percent = array();

    while ($data = mysqli_fetch_array($result)) {

      array_push($co, "CO " . $data['coNum']);
      array_push($percent, $data['percent']);
    }

    ?>


    var co = <?php echo json_encode($co); ?>;
    var percent = <?php echo json_encode($percent); ?>;

    for (var i = 0; i < percent.length; i++) {
      percent[i] = parseFloat(percent[i]);
    }
    document.getElementById("chart-container").innerHTML='';
    document.getElementById("chart-container").innerHTML='<canvas style="background-color:white;height:500px;width:400px;" id="myChart"></canvas>';
    const ctx = document.getElementById('myChart');

    new Chart(ctx, {
      type: 'radar',
      data: {
        labels: co,
        datasets: [{
          label: 'CO Achieved',
          data: percent,
          fill: true,
          backgroundColor: 'rgba(54, 162, 235, 0.2)',
          borderColor: 'rgb(54, 162, 235)',
          pointBackgroundColor: 'rgb(54, 162, 235)',
          pointBorderColor: '#fff',
          pointHoverBackgroundColor: '#fff',
          pointHoverBorderColor: 'rgb(54, 162, 235)'
        }]
      },
      options: {
        elements: {
          line: {
            borderWidth: 3
          }
        }
      }
    });

  }
</script>


</body>

</html>
```

Output Forms cont.



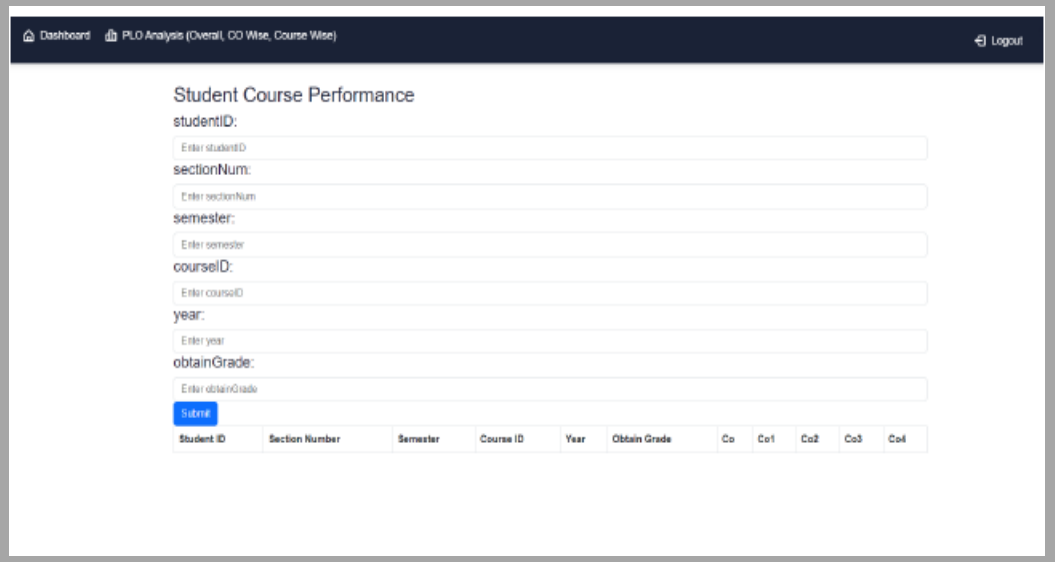
Student Course Performance

Import

studentID	sectionNum	semester	courseID	year	obtainGrade	co	co1	co2	co3	co4
1611001	2	summer	EEE131	2020	B+	80%	80%	80%	80%	80%
1711409	2	autumn	EEE131	2021	A-	85%	85%	85%	85%	85%
1910876	2	autumn	EEE131	2021	A-	85%	85%	85%	85%	85%
1720718	1	summer	EEE231	2020	A-	85%	85%	85%	85%	85%
1722021	1	autumn	ENG101	2021	A	90%	90%	90%	90%	90%
1810471	1	autumn	ENG101	2020	A-	85%	85%	85%	85%	85%
1811135	2	spring	EEE131	2021	B-	70%	70%	70%	70%	70%
1722021	2	spring	EEE131	2021	C	60%	60%	60%	60%	60%
1910876	2	spring	EEE131	2021	C-	55%	55%	55%	55%	55%
1821772	3	spring	ENG101	2020	A	90%	90%	90%	90%	90%
1822089	4	summer	ENG101	2020	B	75%	75%	75%	75%	75%
1910876	1	spring	CSC101	2021	A	90%	90%	90%	90%	90%
1931160	1	spring	CSC101	2021	A	90%	90%	90%	90%	90%
2020076	2	spring	MKT101	2020	A	90%	90%	90%	90%	90%
1711411	2	spring	MKT101	2020	C+	65%	65%	65%	65%	65%
1531176	1	summer	EEE131	2021	C+	65%	65%	65%	65%	65%
1910876	1	summer	EEE131	2021	B+	80%	80%	80%	80%	80%
1531176	1	summer	ENG101	2021	B-	70%	70%	70%	70%	70%

```php
// Get status message
if(!empty($_GET['status'])){
    switch($_GET['status']){
        case 'success':
            $statusType = 'alert-success';
            $statusMsg = 'Data has been imported successfully.';
            break;
        case 'error':
            $statusType = 'alert-danger';
            $statusMsg = 'Some problem occurred, please try again.';
            break;
        case 'invalid_file':
            $statusType = 'alert-danger';
            $statusMsg = 'Please upload a valid CSV file.';
            break;
        default:
            $statusType = '';
            $statusMsg = '';
    }
}
1 reference
function gradeToPercentage($grade){
    switch($grade){
        case 'A': return mt_rand(90, 99);
        case 'A-': return mt_rand(85, 89);
        case 'B+': return mt_rand(80, 84);
        case 'B': return mt_rand(75, 79);
        case 'B-': return mt_rand(70, 74);
        case 'C+': return mt_rand(65, 69);
        case 'C': return mt_rand(60, 64);
        case 'C-': return mt_rand(55, 59);
        case 'D+': return mt_rand(50, 54);
        case 'D': return mt_rand(45, 49);
        default: return mt_rand(1, 44);
    }
}
```

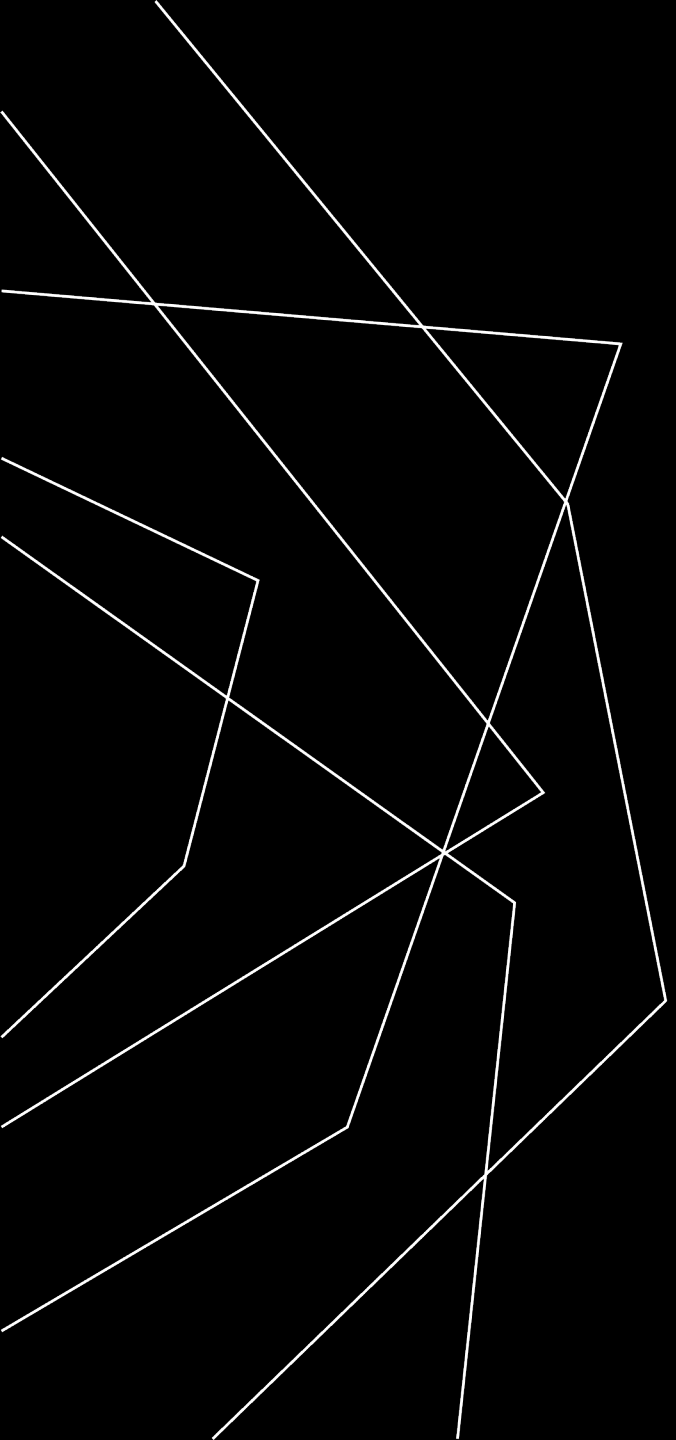
Output Forms cont.




```php
<?php

// Include database connection and header files
require_once(__DIR__."/testfunction.php");
require_once(__DIR__ . "/connect.php");
require_once(__DIR__ . "/user_header.php");
$result = false;
$grade_mapping = [
    "A" => 90,
    "A-" => 85,
    "B+" => 80,
    "B" => 75,
    "B-" => 70,
    "C+" => 65,
    "C" => 60,
    "C-" => 55,
    "D+" => 50,
    "D" => 45,
    "F" => 43,
];
$grade_point_mapping = [
    "A" => 4.0,
    "A-" => 3.7,
    "B+" => 3.3,
    "B" => 3.0,
    "B-" => 2.7,
    "C+" => 2.3,
    "C" => 2.0,
    "C-" => 1.7,
    "D+" => 1.3,
    "D" => 1.0,
    "F" => 0.0,
];


if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if(isset($_POST["studentID"]) && $_POST["sectionNum"] && isset($_POST["semester"]) && isset($_POST["courseID"]) && isset($_POST["year"]) && isset
```

Conclusion

In conclusion, developing the SpmsV4 system posed several challenges throughout the analysis, design, and implementation phases. However, through collaboration with faculty members and stakeholders, we were able to overcome confusion around the system's operations, design challenges, and bugs. Moving forward, integrating extracurricular activity monitoring into the SpmsV4 system could enhance the system's functionality and provide invaluable insights into student engagement for teachers and administrators. To do so, it's important to consider secure data collection and storage, student privacy and consent, and intuitive data presentation. Ultimately, this new feature could help foster a more inclusive and supportive school environment and contribute to students' success both in and out of the classroom.



Thank you

For your time and attention!