

**Database Management Project**

**Spring 2023**

**Final Report of Group-18**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Monjurul Hoque Sharnav | 1821422 |
| Ikterab Yazda | 1910970 |
| Ashir Abdal Ravee | 2020646 |
| S.M. Tanzim Tuhin | 2022302 |
| Tushar Basak | 2022315 |
| Md. Golam Saqlain | 2030404 |
| Shahriar Amin Ronok | 2031361 |

**CONTENTS**

|  |  |
| --- | --- |
| **1** | **INTRODUCTION 2**  **1.1** Background of the Organization **2**  **1.2** Background of the Project **2**  **1.3** Objectives of the Project **3**  **1.4** Scope of the Project **3** |
| **2** | **REQUIREMENT ANALYSIS 4**  **2.1** Rich Picture -> Existing (AS-IS) System **4**  **2.2** Six Element Analysis -> Existing (AS-IS) System **5**  **2.3** Identification and Analysis of problem **14**  **2.4** Rich Picture -> Proposed (TO-BE) System **15**  **2.5** Proposed process and Six Element Analysis **16** |
| **3** | **LOGICAL SYSTEM DESIGN 31**  **3.1** Business Rules **31**  **3.2** Entity Relationship Diagram (ERD) **34**  **3.3** Relational Schema **35**  **3.4** Normalization **36**  **3.5** Data Dictionary **46** |
| **4** | **PHYSICAL SYSTEM DESIGN 66**  **4.1** Input Forms **66**  **4.2** Output Query and Reports **70** |
| **5** | **CONCLUTION 75**  **5.1** Problems and Solutions **75**  **5.2** Additional features and future development **75** |

**1**

**INTRODUCTION**

**1.1** Background of the Organization

Independent University, Bangladesh (IUB) is a private university located in Dhaka, the capital city of Bangladesh. It was founded in 1993 to offer a top-notch education on par with global standards. The university has many departments, including the schools of business, engineering, computer science, liberal arts and social sciences, environmental science and management, and life sciences. In a var, each department provides undergraduate and graduate degree programs in various subjects’ institute for Education and Research, the Center for Sustainable Development, the Institute for Business, Law, and Technology, and the Bangladesh Enterprise Institute are just a few of the research centers and institutions that IUB has to offer in addition to its academic programs.

**1.2** Background of the Project

SpmsV4 system is a software application designed to track and manage the academic progress of students. The system is typically used by teachers, administrators, department Heads, and deans to monitor student performance, identify areas of concern, and track progress over time. Depending on the particular demands and objectives, the project projected backdrop student performance monitoring system may change. A student performance monitoring system can assist in finding students who are having academic difficulty by keeping track of their grades, attendance, and other indicators. This enables instructors and administrators to step in early and offer the assistance the student needs to develop. A performance monitoring system may automate administrative processes like grade reporting and tracking, which lightens the strain on instructors and administrators and enables them to devote more time to education. A student performance monitoring system's overall objective is to guarantee that each student receives the assistance they require to succeed academically.

**1.3** Objectives of the Project

A student performance monitoring system's major goal is to provide a thorough platform for tracking and evaluating students' performance, development, and growth across time. Teachers should be able to enter and record grades for various tasks, tests, and examinations on the system, and students should be able to see their development over time. This system allows instructors and administrators to monitor student attendance to spot trends of absence and take appropriate action. This feature allows instructors and administrators to monitor student attendance to spot trends of absence and take appropriate action.

SpmsV4 keeps relevant documents and data in the database to evaluate the performance of the stakeholders, such as all exam question papers, answer scripts, course outlines, and marks of the examinations and assessments concerning the students' Course Outcomes (CO), Program Learning Outcomes (PLO), and Program Outcomes (PO). Therefore, Students can quantitatively track their progress. Giving them access to a variety of analytical data based on student, course instructor, department, school, and program performance SpmsV4 also gives Higher Authorities chances to draw conclusions and make additional changes.

**1.4** Scope of the Project

While analysis of the existing system (SPMS 3.0), identified some issues in the business processes and ERD which can cause a loss of information in the database. SPMS 3.0 has many functions missing in this project. The proposed solution is to overcome those issues and create a more improved version of the existing system. We are calling this new version SpmsV4 which use a RDBMS (Relational Database Management System) to store, update and retrieve necessary data and generate documents such as Course Outlines, Exam Question and Answer Scripts as well as other necessary required data to monitor student performance and produce other OBE (Outcome-Based Education) reports. It's important to design interface that are intuitive, user-friendly and tailored to each user's specific needs. There are lots of consideration while using for interface design techniques, such as wireframes, mockups and prototypes. The interface is also needed to get feedback from users and iterate designs for interface usability, functionality and collaboration, to ensure that all users can easily access and manipulate the data they need to effectively manage their courses and programs.

**2**

**REQUIREMENT ANALYSIS**

The act of obtaining, comprehending and capturing the needs and expectations of stakeholders for a certain system or piece of software is known as requirement analysis. It entails determining the system's operational or commercial goals, outlining its functional and non-functional needs and creating a set of requirements that the system must follow. The basis for all ensuing activities, including design, development, testing, and deployment, is laid by the requirement analysis phase of the software development lifecycle. The basic requirement analysis starts from the process of capturing requirements in a thorough. Before moving on to the following stage, it is important to check the criteria and get there from all parties.

Examining Requirements: Examining the gathered requirements to make sure they are thorough, consistent, and practical. Requirement analysis assists in ensuring that the finished product satisfies the needs and expectations of the stakeholders and provides commercial value. Early requirement identification and comprehension can assist to prevent costly errors, rework, and delays later in the development process.

**2.1** Rich Picture -> Existing (AS-IS) System

A rich picture is a diagrammatic depiction of a complicated circumstance or issue that aids in understanding the different elements and interconnections at play. It is a technique that's frequently used in organizational analysis and systems thinking to help in understanding and communicating complicated problems.



In this Rich picture stakeholders are:

1. UGC,
2. IBE,
3. Higher Authority,
4. Department,
5. Register Office,
6. Admin,
7. Department Head,
8. Instructor and
9. Student.

The main storage is:

1. Database

**2.2** Six Element Analysis -> Existing (AS-IS) 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 to see student performance pattern and
7. Review and grade change request.

We are using six-element analysis to analyze the impact of six elements in a process. And the six elements are:

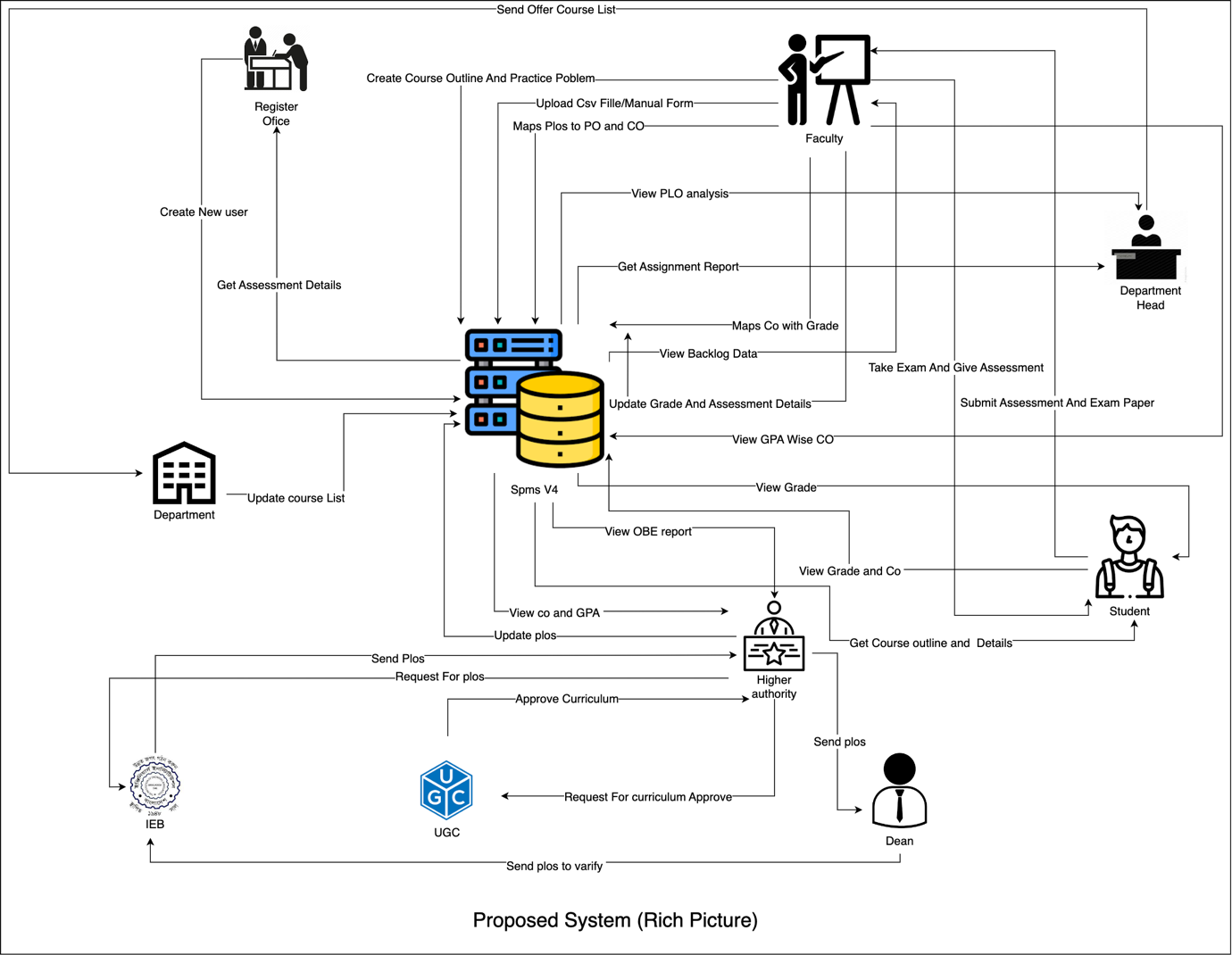
1. Human,
2. Non-Computing Hardware,
3. Computing Hardware,
4. Software,
5. Database and
6. Network and Communications.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Process** | **System Roles** | | | | | |
| **Human** | **Non-Computing Hardware** | **Computing Hardware** | **Software** | **Database** | **Network and Communication** |
| **1. Calculate Course Outcomes (COs) and Program Learning Outcomes (PlOs)** | **A. IBE/UGC/ Higher Authority:**  1. Submit an accreditation requirement containing defined PLOs to the Department office.  **B. Department Office:**  1. Receive an accreditation requirement from IEB.  2. Send the accreditation requirement containing defined PLOs to Faculty Member / Instructor and direct them to design and coordinate courses as instructed.  **C. Faculty Member / Instructor:**  1. List course content and COs.  2. Link CO to PLO and specific questions for examinations and projects.  3. Design a course evaluation report using course content, course outline, and CO. | **A. Pen and Paper:**  1. Used to write down preliminary thoughts and ideas. | **A. Computer:**  1. Used to make softcopies of a course outline, CO.  **B. Printer:**  1. Used to make hard copies of documents such as Course outcomes (CO). | **A. MS Word:**  1. Used to create detailed lesson plans and course assessment reports, mapping course outcome (CO) to program learning outcome (PLO).  **B. Excel Sheet:**  1. Used to outline specific questions in the exam and project to specific Course outcome (CO). |  | **A. Internet and Email:**  1. Used to communicate with UGC / IBE or other topic-related partners to discuss key issues related to mapping Course outcome (CO) and Program learning outcome (PLO).  **B. Others:**  1. For communication with other partners use a phone and physical media to discuss important topics for mapping Course outcome (CO) and Program learning outcome (PLO). |
| **2.**  **Record Student Evaluation Data** | **A. Faculty Member / Instructor:**  1. Assign project work and assignments and also take tests and exams all through the semester.  2. Record student evaluation data throughout the student’s semester for each task (assignment, tests, mid and final exams, project) in both soft and hard copy. And also record marks for each specific question for both mid and final exams.  3. Calculate total marks for assignments, tests, and mid and final exams, and assign final grades to each student for specific courses.  4. Convert midterm and final exam grades and put all of each student’st’st grades for a course into one transcript to classify students and upload student final grades to IRAS.  5. Submit the transcript to the Department office and the Register’s office. | **A. Pen and Paper:**  1. Used to write down data for efficient work. | **A. Computer:**  1. Used to make softcopies of all assessment data for specific courses. | **A. Excel Sheet:**  1. Record the required evaluation data and grades in an Excel Sheet.  **B. IRAS:**  1. Upload the student’s final grade to IRAS for the student or Register’s office to view. | **A. Department Storage:**  1. The record hard copies of student evaluation data and grades retained by the Department and Register’s office for future use.  **B. IRAS Database Server:**  1. IRAS records and maintain students’ grade in the database. | **A. Internet:**  1. Used to access IRAS and upload grades of students. |
| **3. Generate OBE Marksheet and Course Evaluation Report** | **A. Faculty Member / Instructor:**  1. Calculate marks from each task (assignment, tests, mid and final exams, project) to calculate the total mark received for CO.  2. Calculate the total percentage obtained for all CO on the OBE transcript.  3. Report if the student achieves ves spepercentageercent which is greater or equal to 40% and if a student receives a PLO for the CO involved.  4. Prepare a statement and analysis of the number of students who can receive certain CO, PLO, and other documents containing the required information and data.  5. Design a course evaluation report according to the Course outline, Course content, and COs.  6. Submit the final version of the OBE transcript to the Department Office.  **B. Department Office:**  1. Send OBE transcripts, course evaluation reports, and other documents to the Register’s office.  2. Keep OBE transcripts and evaluation reports in the department.  **C. Register’s Office:**  1. Keep OBE transcripts and evaluation reports and other documents in the Register’s office. | **A. Pen and Paper:**  1. OBE transcript save on paper and additional marking may be required for further clarifications. | **A. Computer:**  1. Used to create soft copies of OBE transcript and course evaluation reports.  **B. Printer:**  1. Used to print hard copies of the final version of OBE transcripts and course evaluation reports. | **A. Modified Excel Sheet:**  1. Used to automatically calculate PLO success / failure for students by Faculty Members / Instructors.  **B. MS Word:**  1. Used to make course evaluation report soft copies. | **A. Department Storage:**  1. Hard copies of student evaluation data records and final grads will be kept in the Department for future use.  **B. Register’s Office Storage:**  1. Hard copies of OBE transcript, Course evaluation report and other documents submitted by Department office will be retained for future use. | **A. Internet:**  1. An online platform such as google docs, google sheet can be used to process the OBE evaluation data. |
| **4.**  **View result and download Transcript** | **A. Student:**  1. Login to IRAS.  2. Search semester wise result for expected semester and see grades.  3. Download the transcript via browser into drive.  **B. Register’s Office:**  1. Access IRAS and view students’ grades if needed.  2. Download the transcript via browser into drive. | **A. Pen and Paper:**  1.Transcripet can be printed in paper and maybe required for additional marking. | **A. Computer:**  1. Used for accessing IRAS.  **B. Printer:**  1. Used for print transcript. | **A. IRAS:**  1. Keep grade for each course and gives the online client interface for seeing grades and transcripts. | **A. Register’s Office Storage:**  1. Student information is kept in hardcopies for future use.  **B. IRAS Database Server:**  1. Used to keep, maintain, receive and modify student grade and other information.  **C. Remote Database Server:**  1. For accessing user interface and web pages saved by remote web server. | **A. Internet and Email:**  1. Used to connect with IRAS and mailing softcopies of documents. |
| **5.**  **Create Student/ Faculty/ Staff accounts and insert /adjust necessary data** | **A. Admin:**  1. Information about new students is collected from the registration process.  2. Information about new faculty member is collected from HR.  3. Create accounts for students, faculty and also able to adjust account details as needed. | **A. Pen and Paper:**  1. Can be used to copy / write important credentials to create an account. | **A. Computer:**  1. Used to access and add / modify data in IRAS. | **A. IRAS:**  1. Used as client interface to connect with student/faculty information. | **A. Register’s Office Storage:**  1. Student / Faculty information is kept in hard copy for future use.  **B. IRAS Database Server:**  1. Used to keep, maintain, receive, and modify student and faculty information.  **C. Remote Database Server:**  1. For accessing user interface and web pages saved by the remote web server. | **A. Internet:**  1. Used to connect with IRAS and remote database server. |
| **6.**  **Analyze student’s records, OBE Marksheets and Course evaluation reports over time to see student performance pattern** | **A. IBE / UGC:**  1. Notify university administrators of a due date inside which OBE transcript, course evaluation report and other documents are required for quality review to make the necessary improvements to any programs and if the government officials will be visiting the campus.  2. Visit any relevant department in the university to obtain required documents and reports.  **B. Department:**  1. Department will collect the required documents, OBE transcript, course evaluation reports within the specified time by IBE / UGC.  2. Give all the essential documents to government authorities (IBE / UGC) and VC of the university.  3. Find the issue based on student performance pattern where instructional resources can be modified or improved.  **B. Office of VC:**  1. Based on request, review OBE transcript, course evaluation reports and student performance pattern. | **A. Pen and Paper:**  1. Used to record / note key points from the report and hard copies of the report is also used. | **A. Computer:**  1. Used to view and share soft copies of OBE transcript, course evaluation report and other required documents. |  | **A. Department Storage:**  1. Used to keep, maintain and to retrieve hard copies of OBE transcript, course evaluation report, student performance pattern and other required documents. | **A. Internet and Email:**  1. An online platform such as google docs, google sheet can be used to view reports and email can be used for sharing reports. |
| **7.**  **Review and grade change request** | **A. Student:**  1. Request for review and change of grade to the faculty member / instructor.  **B. Faculty Member / Instructor:**  1. Review answer sheets, other tasks upon request and if request is valid then send a request to admin for grade change but if request is not valid then no request to admin is needed.  **C. Admin:**  1. Modify student grade based on instructor request. | **A. Pen and Paper:**  Used to record / note key points and marking in the hard copy of the answer sheet. | **A. Computer:**  1. Used for communication between student, faculty and admin. | **A. IRAS:**  1. Admin uses IRAS for grade change. | **A. IRAS Database Server:**  1. Used to update new grade.  **B. Department Storage:**  1. Used to update new grade in hard copy. | **A. Internet, Email and Phone:**  1. Email and phone is basically used for communication. |

**2.3** Identification and Analysis of problem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Process Name** | **Stockholders** | **Problems** | **Analysis** | **Proposed Solution** |
| **Record student evaluation data** | **1.** Admin  **2.** Faculty member  **3.** Student | **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 may be 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.** Admin  **2.** Faculty member  **3.** Student | **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 interface to view and print them. |

**2.4** Rich Picture -> Proposed (TO-BE) System

****

In this Rich picture stakeholders are:

1. UGC,
2. IBE,
3. Higher Authority,
4. Department,
5. Register Office,
6. Dean,
7. Department Head,
8. Faculty and
9. Student.

The main storage is:

1. Spms V4.

**2.5** 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.

We are using six-element analysis to analyze the impact of six elements in a process. And the six elements are:

1. Human,
2. Non-Computing Hardware,
3. Computing Hardware,
4. Software,
5. Database and
6. Network and Communications.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **process** | **System Roles** | | | | | |
| **Human** | **Non-Computing Hardware** | **Computing Hardware** | **Software** | **Database** | **Network and Communication** |
| **1.**  **Make and deliver course outlines and lesson plans** | **A. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for course outline and lesson plan.  3. Select option for upload and will submit necessary data.  4. After clicking submit button, inserted data will be uploaded to database.  **B. Student:**  1. Log in to SpmsV4 with username and password.  2. Select option for course outline and lesson plan.  3. Select option for course and course outline and lesson plan will be shown with a download option. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of course outline on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student grade and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **2.**  **Statistics of student enrollment under VC / Dean / Head of Department supervision** | **A. VC:**  1. Log in to SpmsV4 with username and password.  2. Select option for enrollment.  3. Select option for semester and educational year and a submit button.  4. After clicking submit button, statistics of data will be shown.  **B. Dean:**  1. Log in to SpmsV4 with username and password.  2. Select option for enrollment.  3. Select option for semester and educational year and a submit button.  4. After clicking submit button, statistics of data will be shown.  **C. Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for enrollment.  3. Select option for semester and educational year and a submit button.  4. After clicking submit button, statistics of data will be shown. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of student enrollment data on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student grade and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **3.**  **Include question sheets of quizzes, assignments and exams and evaluate replays of answer sheet** | **A. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for question sheets.  3. Select option for course and section and add question sheet and click assign button.  4. And for gradings select option for evaluation and submit marks.  **B. Student:**  1. Log in to SpmsV4 with username and password.  2. Select option for assigned task.  3. Select any task and upload answer and click submit button.  4. And for checking gradings select option for grade and also select course and exams. Grade will be shown. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of question sheets and grade sheets on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student grade and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **4.**  **Statistics of student performance patterns by GPA within any course** | **A. VC / Dean / Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for course student performance. All the courses under users’ supervision will be shown.  3. Select any course or all courses and click submit button.  4. Statistics of selected courses will be displayed graphically.  **B. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for course student performance. All the courses under users’ supervision will be shown.  3. Select any course and click submit button.  4. Statistics of selected course will be displayed graphically. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of student performance data sheets of any course on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student grade and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **5.**  **Statistics of student performance patterns by GPA under faculty supervision** | **A. VC / Dean / Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for faculty student performance. All the courses under selected faculties supervision will be shown.  3. Select any course or all courses and click submit button.  4. Statistics of selected courses will be displayed graphically.  **B. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for faculty student performance. All the courses under users’ supervision will be shown.  3. Select any course and click submit button.  4. Statistics of selected course will be displayed graphically. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of student performance data sheets under faculty supervision on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student grade and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **6.**  **Statistics of student performance patterns by GPA under VC / Dean / Head of Department supervision** | **A. VC / Dean / Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for student performance. All the courses under users’ supervision will be shown.  3. Select any course or all courses and click submit button.  4. Statistics of selected courses will be displayed graphically. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of student performance data sheets under VC / Dean / Head of Department supervision on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student grade and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **7.**  **Course Outcome (CO) and Program Outcome (PO)** | **A. VC / Dean / Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for CO / PO and select, semester, course / program and click submit button.  3. CO / PO earned by students from that course / program will be shown graphically.  **B. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for CO / PO and all the courses of user supervision will be shown course / program and click submit button.  3. CO / PO earned by students from that course / program will be shown graphically.  **C. Student:**  1. Log in to SpmsV4 with username and password.  2. Select option for CO / PO.  3. All CO / PO that student achieved will be displayed graphically. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of student CO / PO reports on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student CO / PO’s atomic data and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **8.**  **Program Learning Outcome (PLO)** | **A. VC / Dean / Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for PLO and select, semester, course / program and click submit button.  3. PLO earned by students from that course / program will be shown graphically.  **B. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for PLO and all the courses of user supervision will be shown course / program and click submit button.  3. PLO earned by students from that course / program will be shown graphically.  **C. Student:**  1. Log in to SpmsV4 with username and password.  2. Select option for PLO.  3. All PLO that student achieved will be displayed graphically. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of student PLO reports on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student PLO’s atomic data and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **9.**  **Statistics of Courses, Programs, Departments and Schools CLO and PLO** | **A. VC / Dean / Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for CLO / PLO and select, course or program or department or school and click submit button.  3. Statistics of selected options data will be displayed graphically.  **B. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for CLO / PLO and all the courses of user supervision will be shown course / program and click submit button. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of reports on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student CLO / PLO’s atomic data and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |
| **10.**  **Statistics of average CLO / PLO of students under Department Supervisions of total number of attempts** | **A. VC / Dean / Head of Department:**  1. Log in to SpmsV4 with username and password.  2. Select option for department wise CLO / PLO and select semester, department and click submit button.  3. Statistics of selected semester of a department will be displayed graphically.  **B. Faculty:**  1. Log in to SpmsV4 with username and password.  2. Select option for department wise CLO / PLO and select semester and click submit button.  3. Statistics of selected semester of a department will be displayed graphically.  **C. Student:**  1. Log in to SpmsV4 with username and password.  2. Select option for department wise CLO / PLO and select semester and click submit button.  3. Statistics of selected semester of a department will be displayed graphically. |  | **A. Computer:**  1. Used to log in to SpmsV4.  **B. Printer:**  1. Used to print hard copies of student CLO / PLO reports on demand. | **A. SpmsV4:**  1. Used as an interface to work with data in the SpmsV4 database. | **A. SpmsV4 database server:**  1. Used to keep, maintain, modify and use student CLO / PLO’s atomic data and other information by system admin. | **A. Internet:**  1. Used to access SpmsV4. |

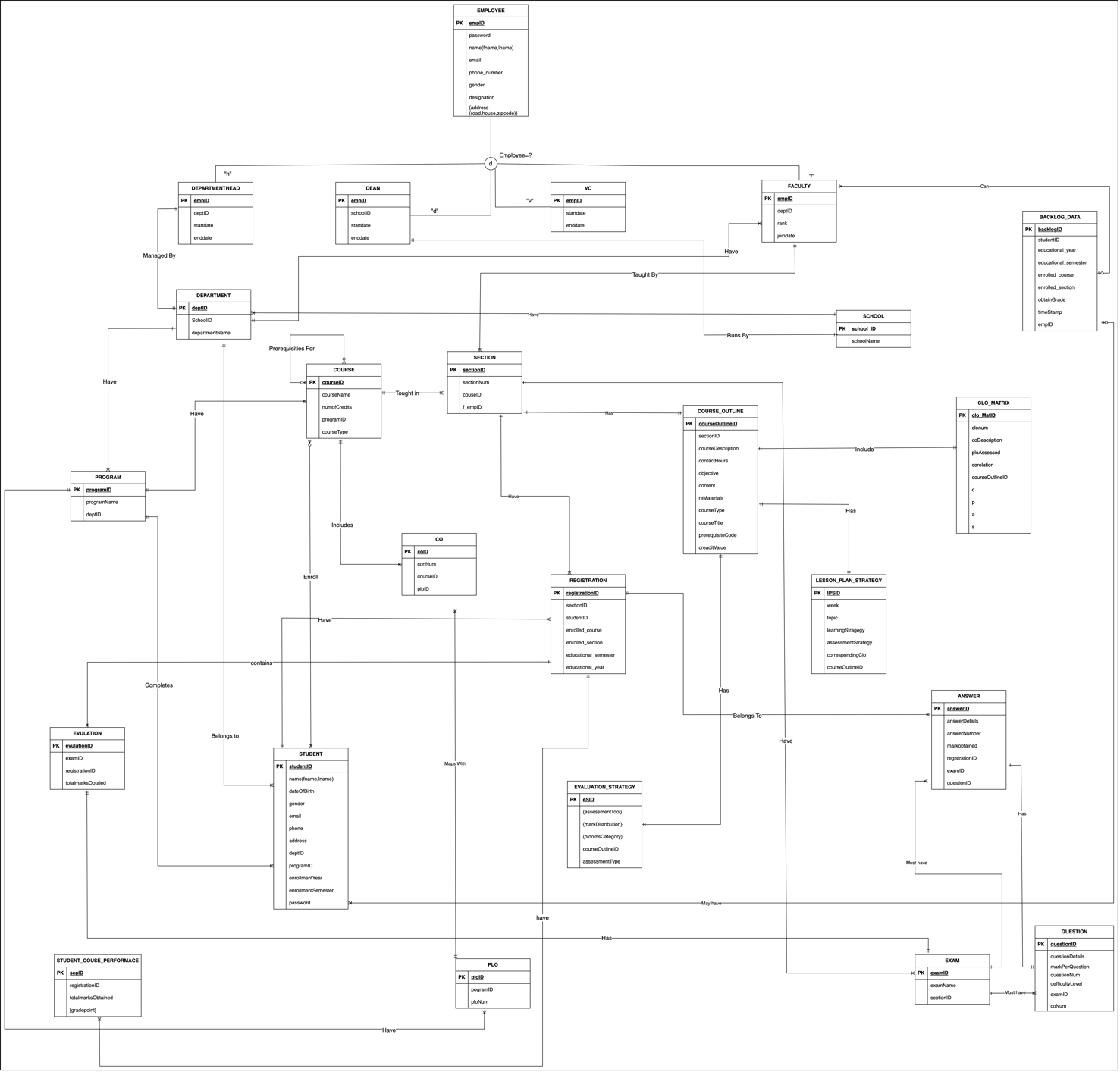
**3**

**LOGICAL SYSTEM DESIGN**

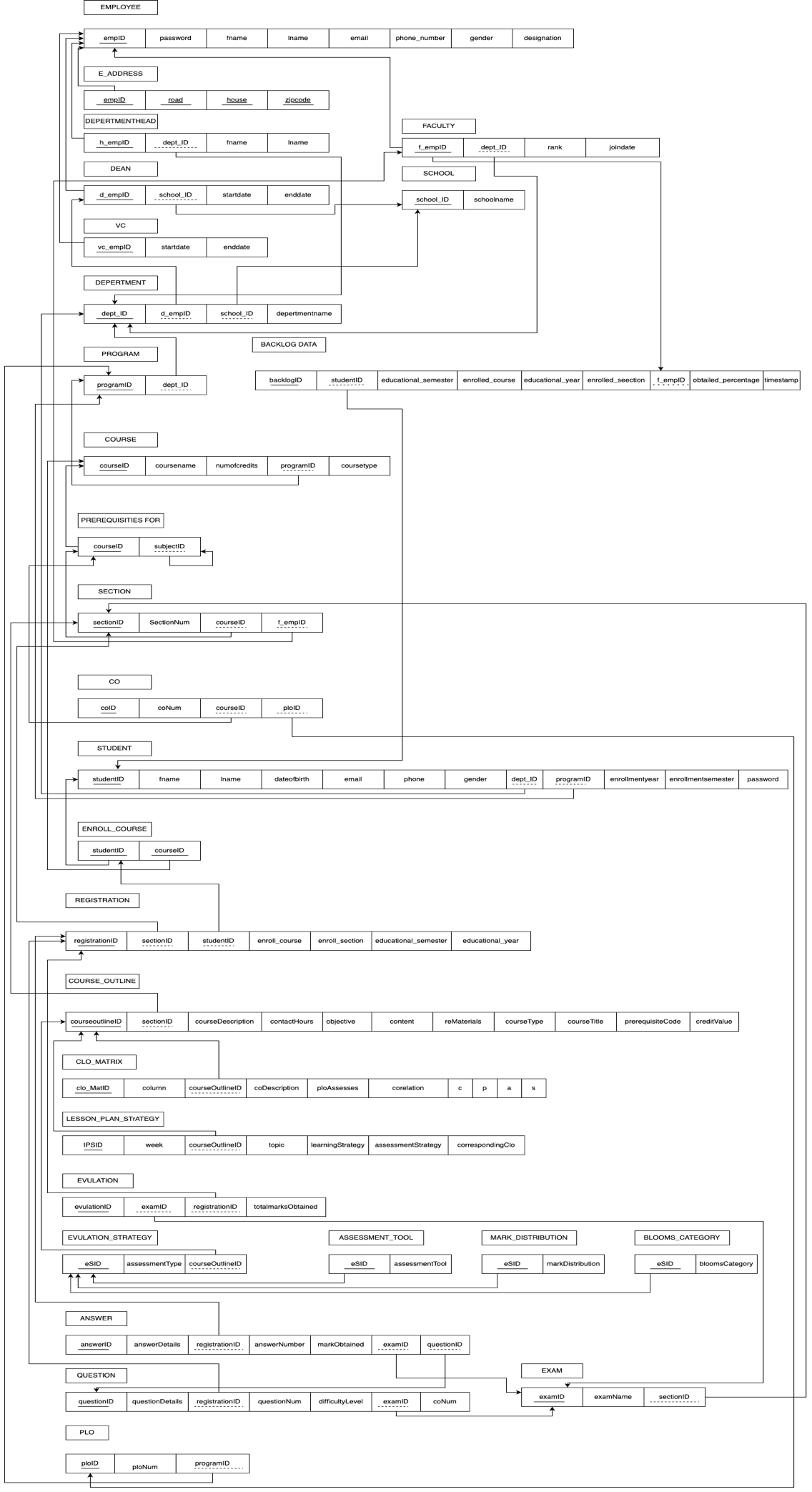
**3.1** Business Rules – SpmsV4

1. A student must have one department. A STUDENT has studentID, name consisting of fname and lname, dateOfBirth, gender, email, phone, address, deptID, programID, enrollmentYear, enrollmentSemester, and password. A department must have one or many students.
2. A student may perform many registrations. A REGISTRATION includes registrationID, sectionID, studentID, enrolled\_course, enrolled\_section, educational\_semester, educational\_year. Registration must be performed by exactly one student.
3. A student must complete exactly one program. A PROGRAM has programID, programName, deptID. A program must be completed by one or many students.
4. A section mandatorily has many registrations. A registration must have at least one section. A Section includes sectionID, sectionNum, courseID, f\_empID.
5. A course must have one program. A program must contain one or many courses. A COURSE contains courseID, courseName, numofCredits, courseType, programID. Many courses are perquisites for many courses. A course must be taught in one or many sections. A section must be taught by exactly one course.
6. A course must include one or many CO. A CO has coID, coNum, courseID, ploID. A CO must be included in exactly one course.
7. A registration may belong to many evaluations. An evaluation mandatorily belongs to one registration. An EVALUATION contains evaluationID, examID, registrationID, totalmarksObtained.
8. A CO must map with one PLO. A PLO's must map with one or many CO's. PLO includes ploID, ploNum, programID. A PLO must contain one program but a program must contain one or many PLO's.
9. A program must belong to one department. A department must belong to one or many programs. A DEPARTMENT contains deptID, departmentName, schoolID.
10. A department must contain one school. A school must contain one or many departments. A SCHOOL includes school\_ID, schoolName.
11. An employee has four sub-types with Department Head, Dean, VC, Faculty. An EMPLOYEE includes empID, password, name consist of fname and lname, email, phone\_number, gender, designation, addresses with road, house and zip code number.
12. A school must be run by exactly one. A dean must run exactly one school. A DEAN has empID, schoolID, startdate, enddate.
13. A Department must be run by exactly one Department head. A department head must manage exactly one department. A DEPARTMENTHEAD includes empID, deptID, startdate, enddate.
14. A Faculty must have exactly one Department. A department must have one or many Faculties. A FACULTY includes empID, deptID, rank, joindate. A faculty may teach many sections. A section must be taught by exactly one faculty.
15. A course outline belongs to exactly one section. A section must have exactly one course outline. A COURSE\_OUTLINE includes courseOutlineID, sectionID, contactHours, courseDescription, objective, content, refMaterials, courseType, courseTitle, prerequisiteCode, creditValue.
16. A course outline must have exactly one CLO Matrix. A CLO matrix belongs to exactly one course outline. A CLO\_MATRIX includes clo\_MatID, cloNum, coDescription, ploAssessed, correlation, courseOutlineID, c, p, a, s.
17. A Lesson Plan Strategy must have exactly one Evaluation strategy. An Evaluation strategy must have exactly one Lesson Plan Strategy. A LESSON\_PLAN\_STRATEGY includes lPSID, week, topic, learning Strategy, assessmentStrategy, correspondingClo, courseOutlineID.
18. An exam has exactly one evaluation. An Evaluation for an exam is done exactly once. An EXAM includes examID, examName, sectionID. A section must have one or many exams. An exam must be held in one section.
19. An exam must have one or many questions. Every question must belong to exactly one exam. A QUESTION includes questionID, questionDetails, markPerQuestion, questionNum, difficutltyLevel, examID, coNum. A Question is answered exactly once. An answer has exactly one question.
20. An answer must have one exam. An ANSWER includes answerID, answerDetails, answerNumber, markobtained, registrationID, examID, questionID. An exam must have multiple answers. A registration must have many answers but an answer has one registration.
21. A student course performance evaluation is done for registration exactly once. A registration must have many students course performance evaluation. A STUDENT\_COURSE\_PERFORMANCE includes scpID, registrationID, totalmarksObtained and gradepoint that is calculated from grade.
22. A faculty may enter backlog data only once. A backlog data must be entered by many faculties. A BACKLOG\_DATA includes backlogID, studentID, educational\_year, educational\_semester, enrolled\_course, enrolled\_section, obtained\_percentage, timestamp, empID. Every backlog data must need one or many students. A student may have one or many backlog data.

**3.2** Entity Relationship Diagram (ERD)



**3.3** Relational Schema



**3.4** 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 |
| e1 | e2, 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.

R10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 |

R11

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| l1 | 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 |

R12

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | a3 | a4 | r1 | r2 | r3 | r4 | r5 | ex1 |
| ex2 | s1 | s2 | s3 | s4 | s5 | s6 | s7 | s8 | s9 |
| s10 | q1 | q2 | q3 | q4 | q5 | q6 |

R13 R14

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| es1 | es2 | es3 | es4 | es5 |  |  |  | ev1 | ev2 |

PK

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e1 | l1 | a1 | es1 | ev1 | cm1 | co1 | sp1 |

R15

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cm1 | cm2 | cm3 | cm4 | cm5 | cm6 | cm7 | cm8 | cm9 |

R16 R17

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| co1 | co2 | pl1 | pl2 |  |  |  | sp1 | sp2 |

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

R10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 |

R111 R113

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| l1 | l2 | l3 | l4 | l5 | l6 | cu1 |  | st1 | st2 | c1 | f1 |

R112

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cu1 | cu2 | cu3 | cu4 | cu5 | cu6 | cu7 | cu8 | cu9 | cu10 | st1 |

R114 R115

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| c1 | c2 | c3 | c4 | p1 |  | f1 | f2 | f3 | d1 |

R116 R118 R117

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p1 | p2 |  | sc1 | sc2 |  | d1 | d2 | sc1 |  |

PK

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| e1 | l1 | a1 | es1 | ev1 | cm1 | co1 | sp1 |

R121

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | a3 | a4 | r1 | q1 | ex1 |

R122 R125

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| r1 | r2 | r3 | r4 | r5 | s1 |  | q1 | q2 | q3 | q4 | q5 | q6 |

R124

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s1 | s2 | s3 | s4 | s5 | s6 | s7 | s8 | s9 | s10 |

R123 R13

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ex1 | ex2 |  | es1 | es2 | es3 | es4 | es5 |

R14 R15

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ev1 | ev2 |  | cm1 | cm2 | cm3 | cm4 | cm5 | cm6 | cm7 | cm8 | cm9 |

R161 R162 R17

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| co1 | co2 | pl1 |  | pl1 | pl2 |  | sp1 | sp2 |

**3.5** Data Dictionary

Employee\_T

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Size | Remark |
| nempIoyeeID | 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”. |

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\_empIoyeeID | 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\_empIoyeeID | 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” |

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” |

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 |
|  | | | |

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” |

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 |

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 |

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. |

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” |

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” |

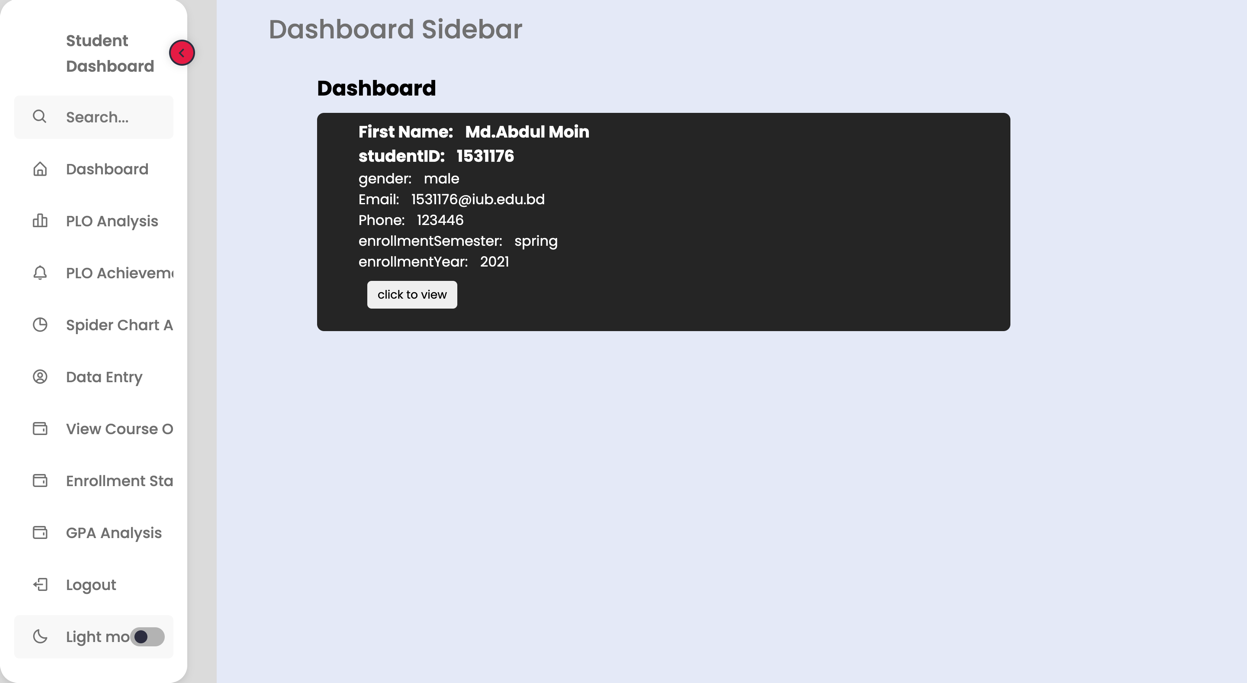
**4**

**PHYSICAL SYSTEM DESIGN**

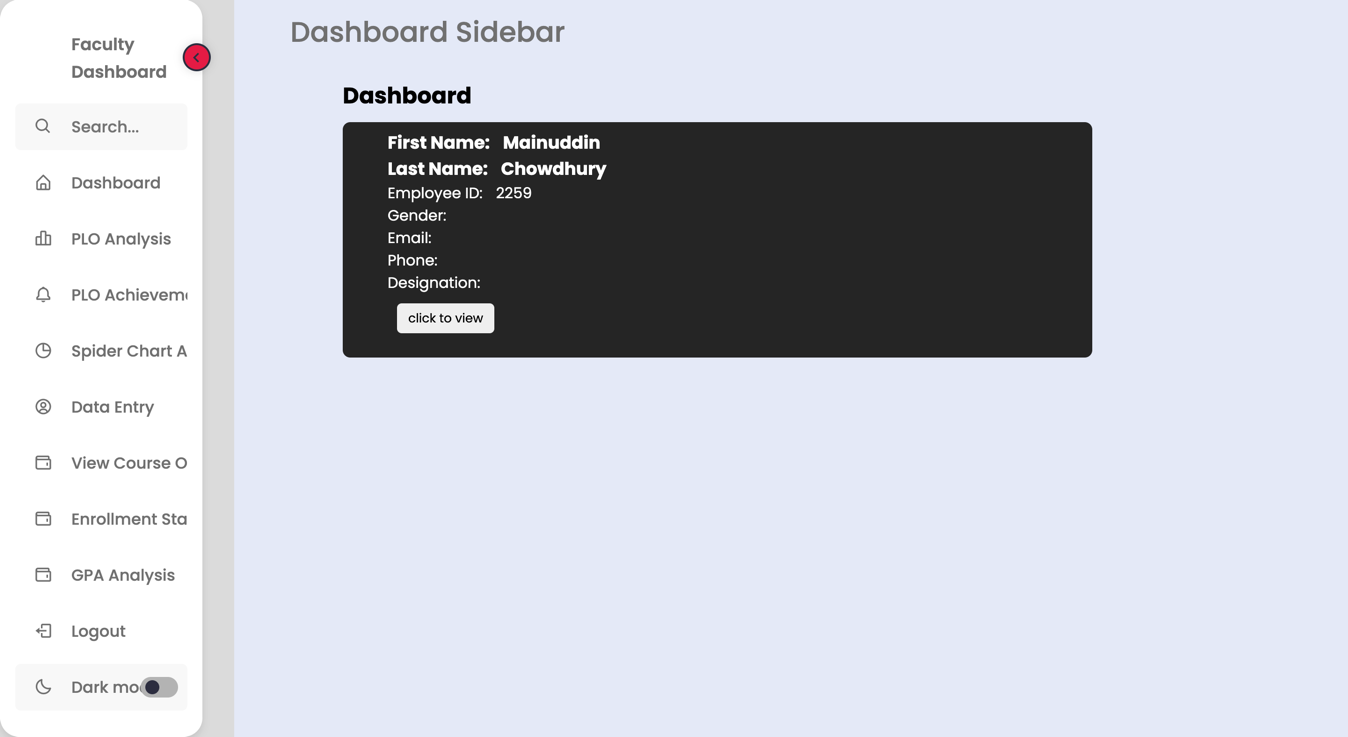
**4.1** Input Forms

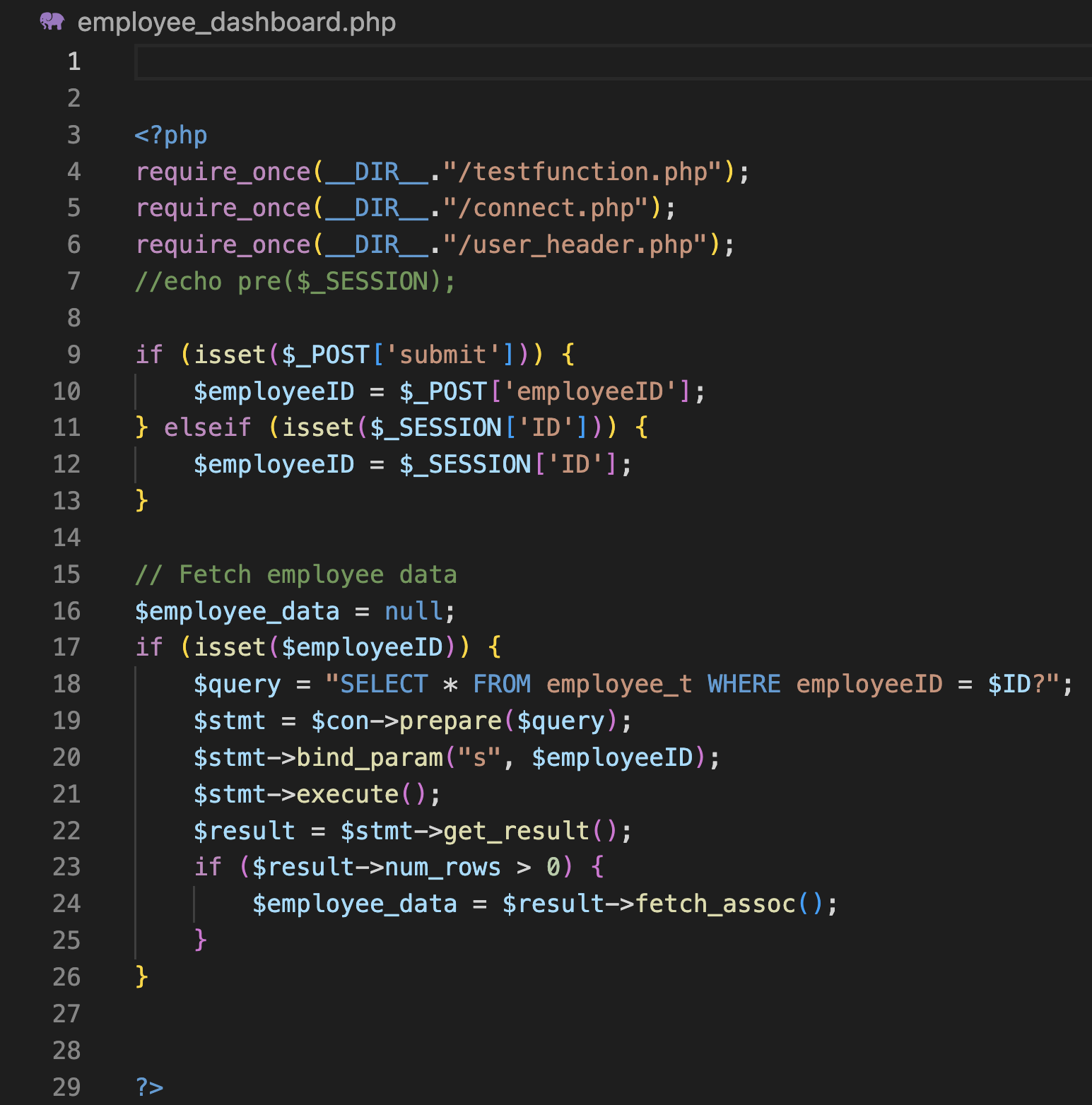










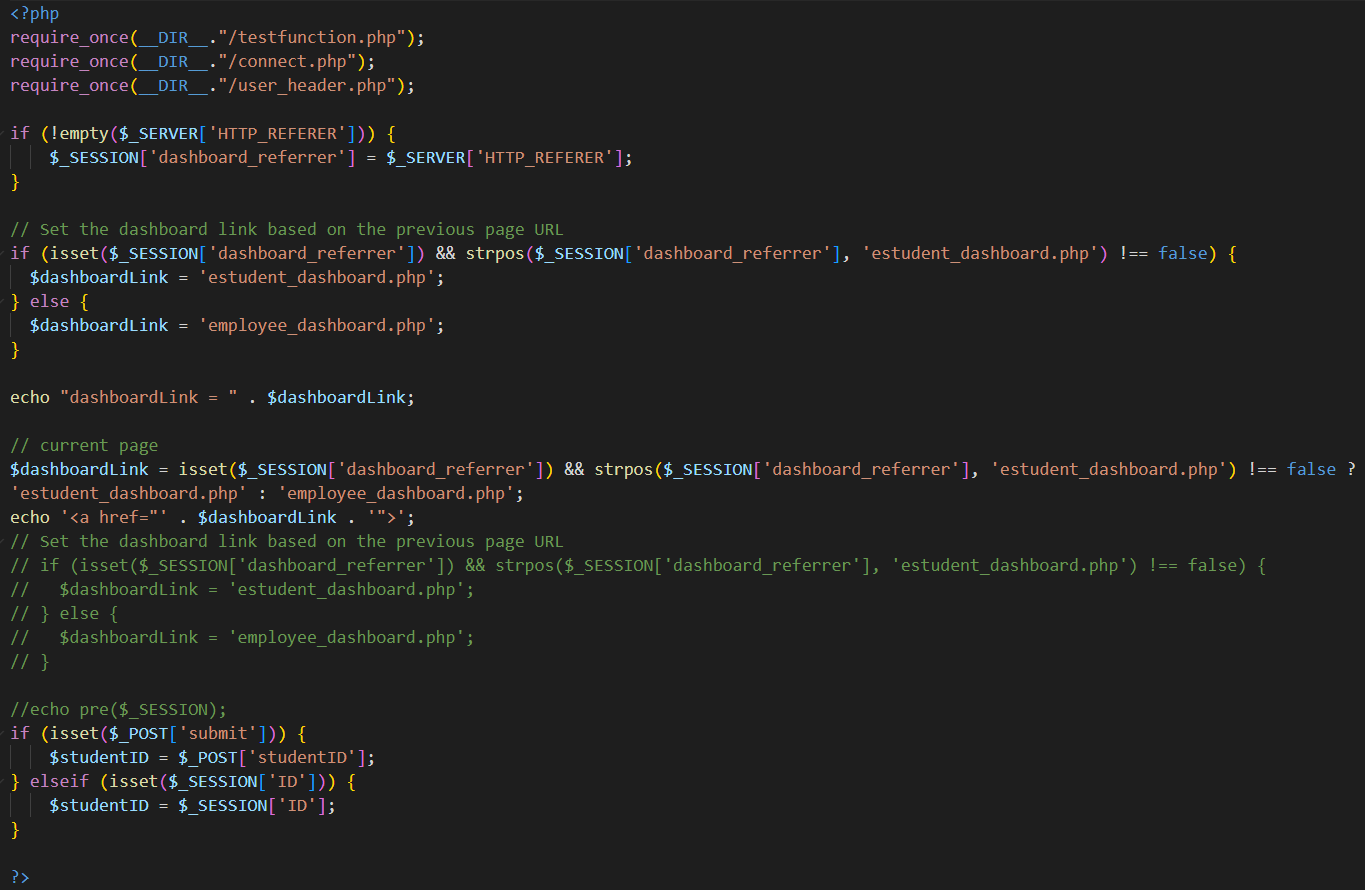


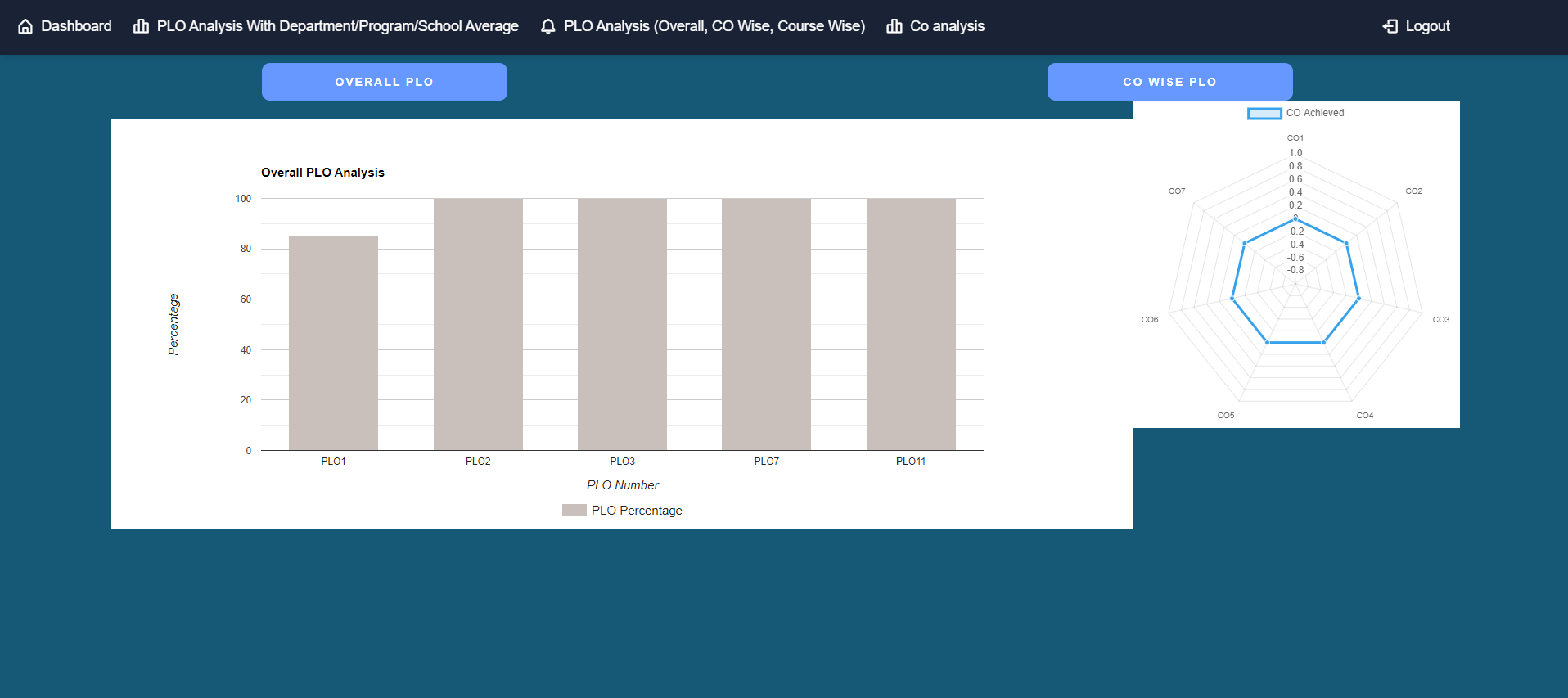




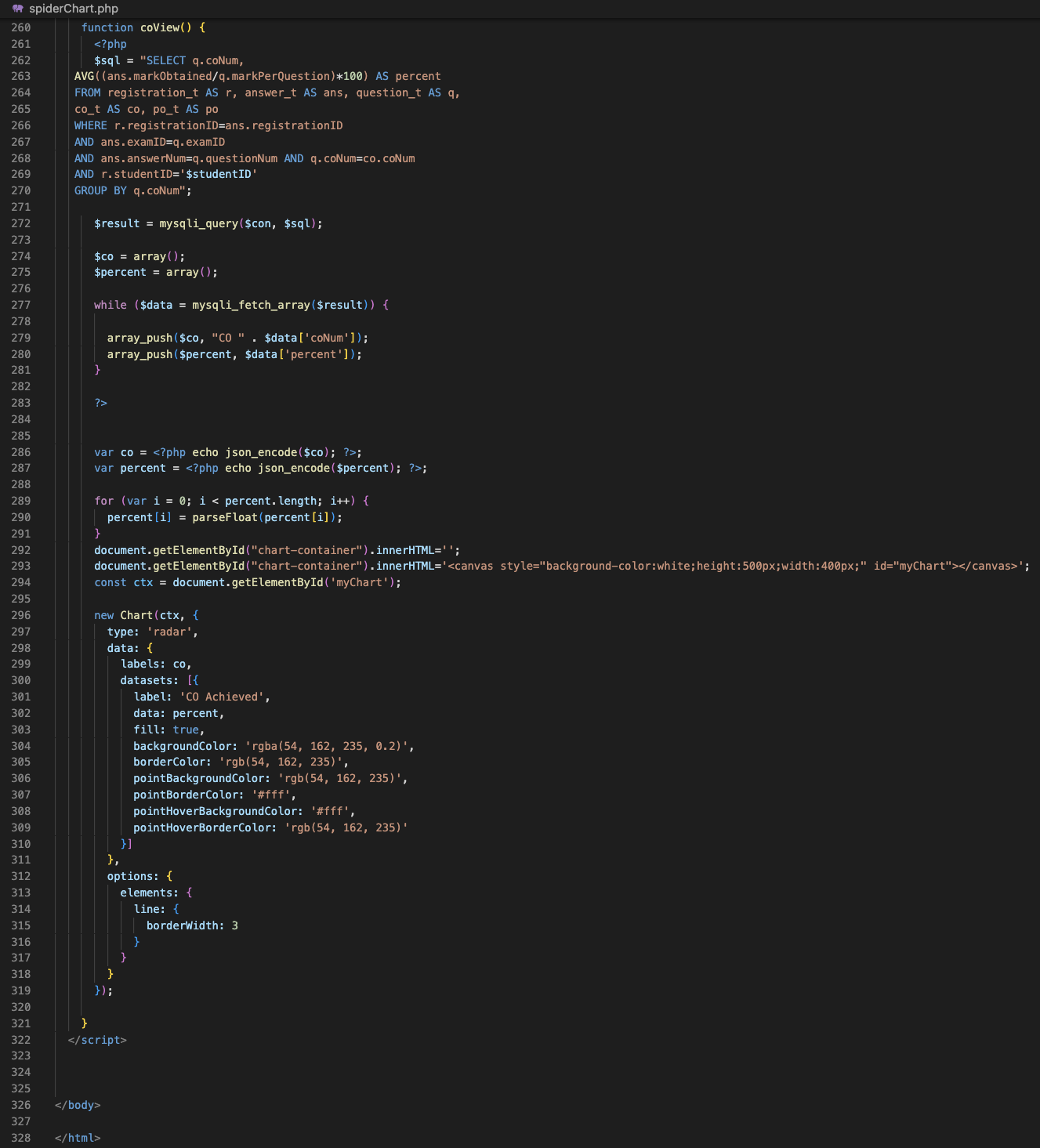
**4.2** Output Query and Reports

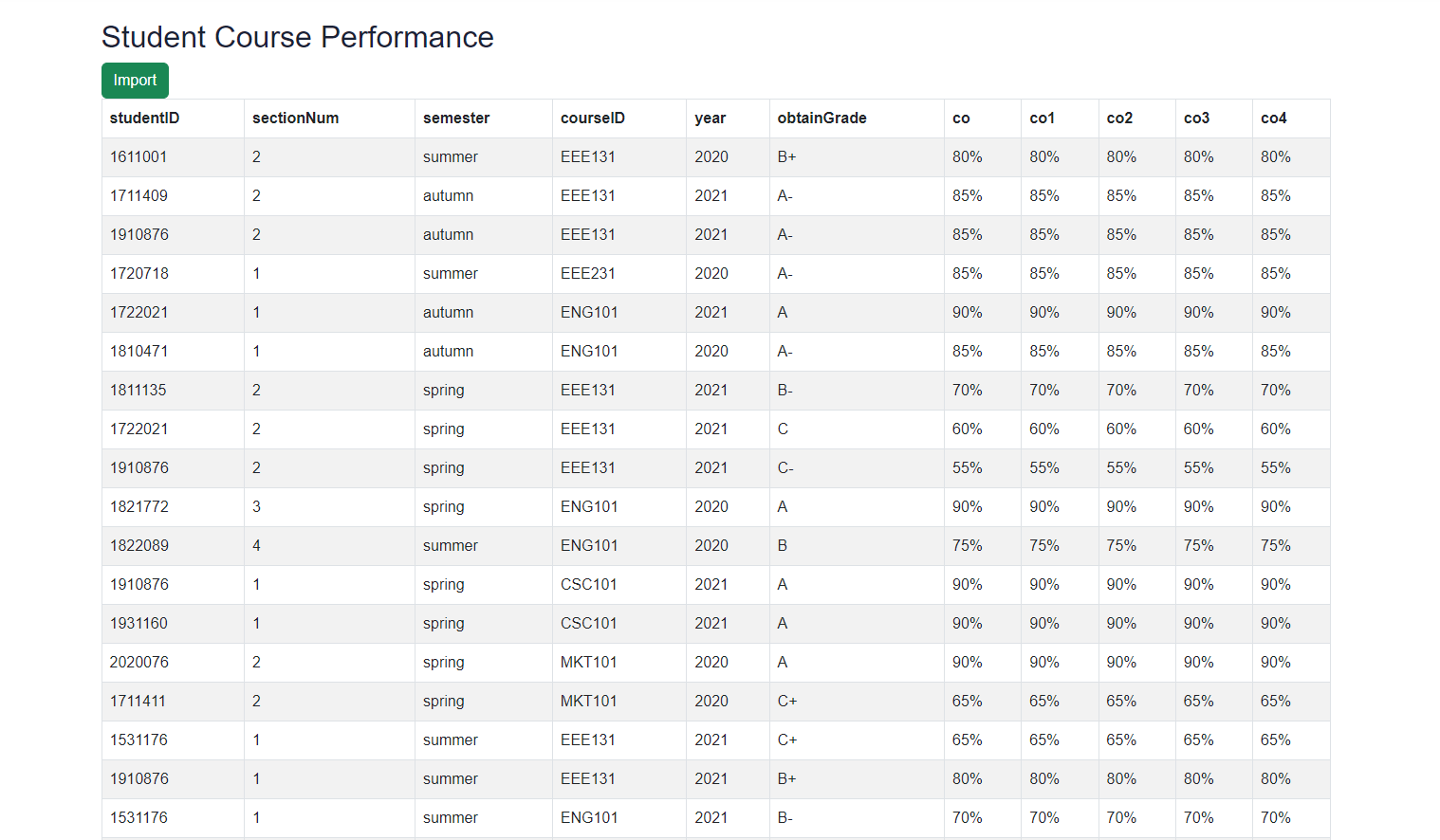


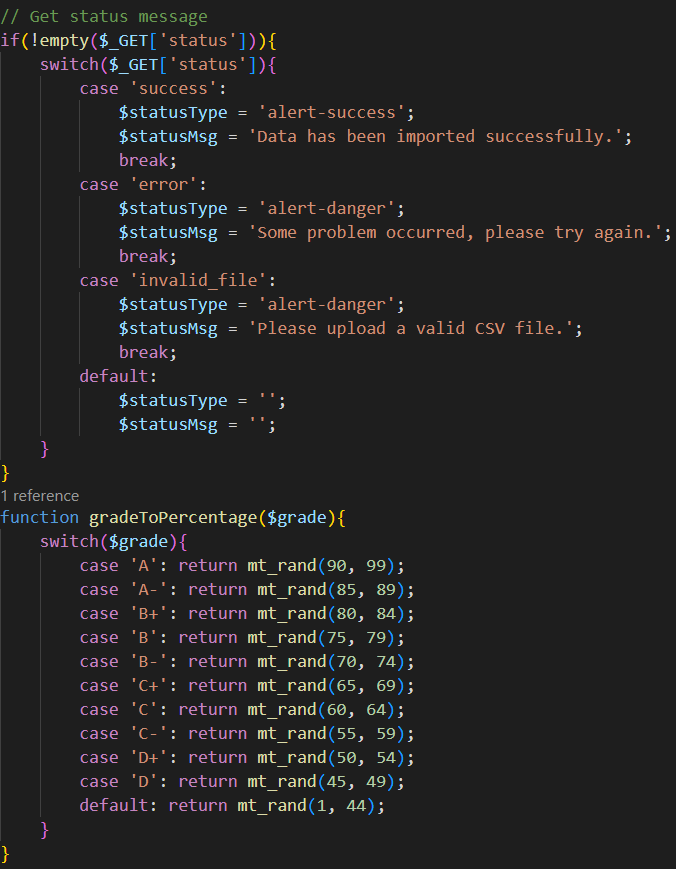


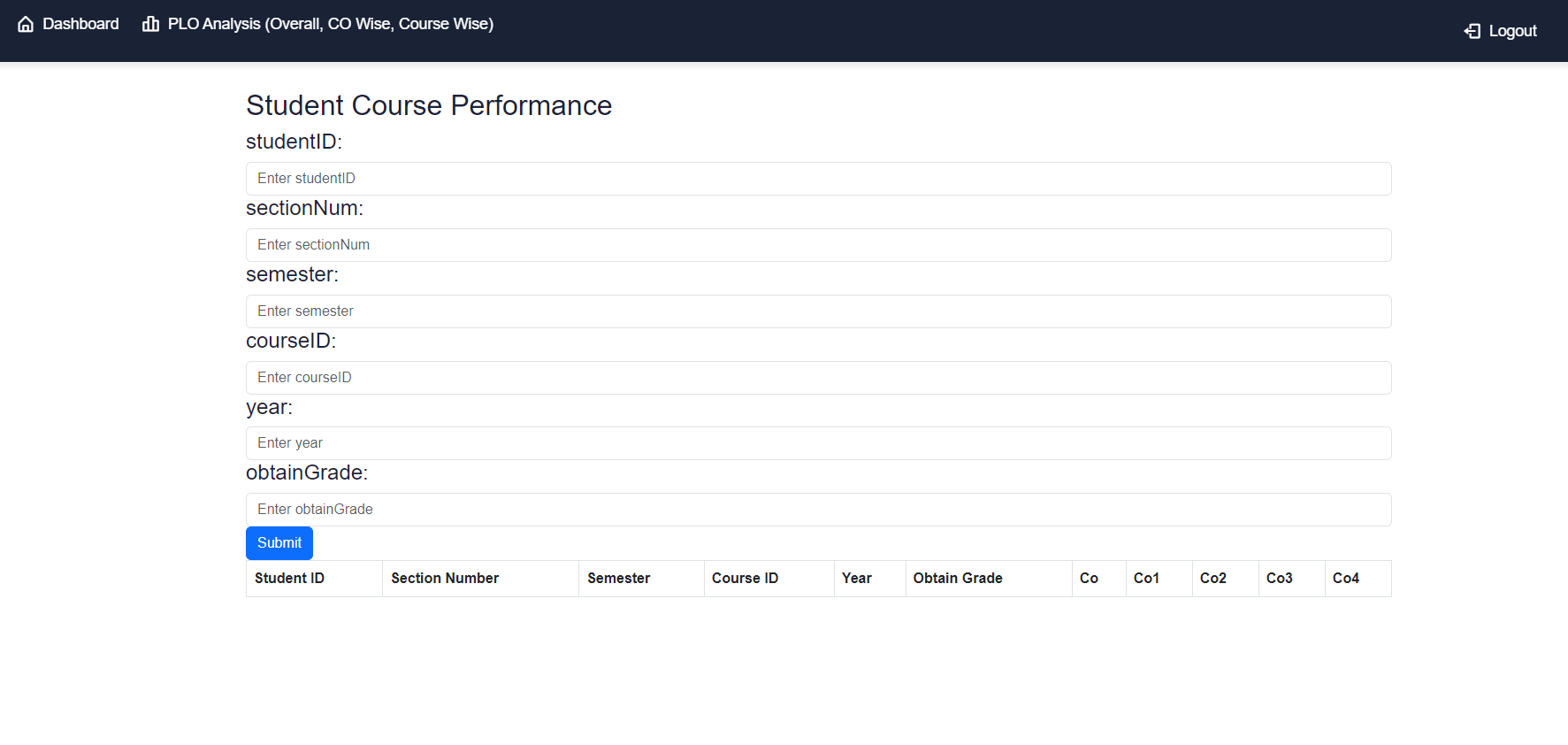


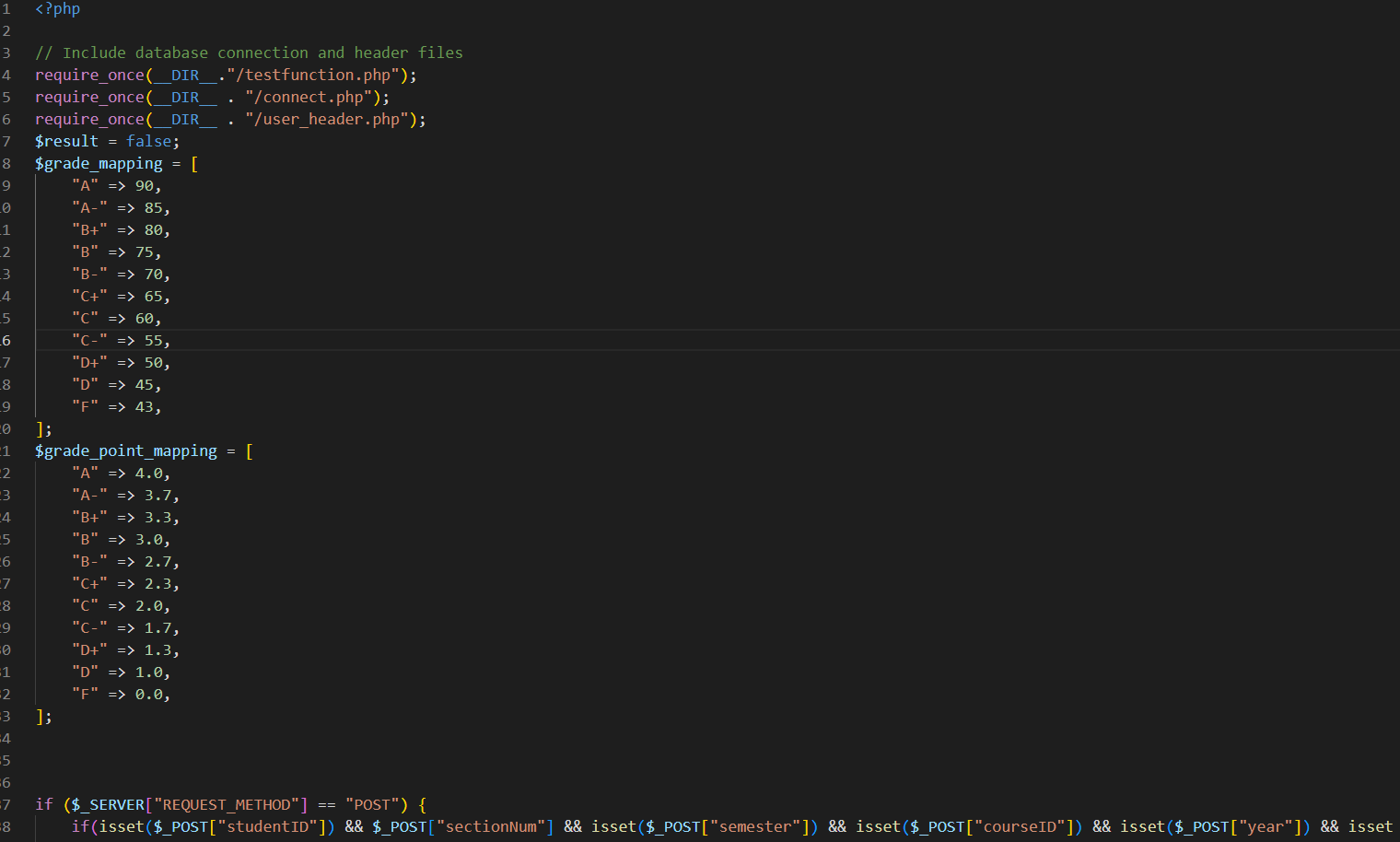






****

****

****

**5**

**CONCLUTION**

**5.1** Problems and Solutions

* Analysis Phase:

One of the major problems faced was the confusion around the Rich Picture and Six Element Analysis of the organizational operations since there was no data available regarding those operations. However, Faculty members and other stakeholders were interviewed to overcome such confusion, and information received during the interview was collected to get a better understanding of the system that was being developed.

* Design Phase:

We face many serious problems during the design phase and ERD. Many systems and calculations were missing in the program. The front end didn't show what was expected. the dashboard was the same student and faculty. Somehow, we manage to overcome all those problems.

* Implementation Phase:

We try to implement all the new requirements and update all previous bugs. For that implementation, we used –

Front-End Developing tools: HTML, CSS, JavaScript, Google Charts, Chart JS Back-End Developing tools: PHP, JSON

Database-integration: MySQL.

**5.2** Additional features and future development

By integrating extracurricular activity monitoring into the SpmsV4 system, you could enable students to easily log and track their participation in various school activities, clubs, and events. This could include sports teams, academic clubs, music programs, volunteer work, and more. The system could then generate reports and analytics that highlight the student's level of participation, leadership roles, achievements, and areas for improvement.

From an educator's perspective, this data could be invaluable for gaining a holistic view of each student's engagement in school life. Teachers and administrators could use this information to identify students who may need additional support or encouragement or to recognize those who have demonstrated exceptional leadership and teamwork skills. This could ultimately help to foster a more inclusive and supportive school environment.

To implement this feature, you would need to consider how to securely collect and store extracurricular activity data, how to ensure privacy and consent from students, and how to present the data in an intuitive and meaningful way. You may also want to consider integrating the feature with existing systems, such as student profiles and attendance tracking, to provide a more complete picture of each student's performance and progress.

**Reference:**

1. [**http://www.iub.edu.bd/AboutIUB/ataglance**](http://www.iub.edu.bd/AboutIUB/ataglance)
2. [**http://www.cse.iub.edu.bd/degrees/1**](http://www.cse.iub.edu.bd/degrees/1)
3. [**https://www.youtube.com/watch?v=neUOwhzLa0w**](https://www.youtube.com/watch?v=neUOwhzLa0w)

**THE END**