Database Management Project

Final Report

Group No. 34

Group Name: MASHA ALLAH

Name	ID	Theory Section	Lab Section
Sanjidul Alam Chowdhury	2120394	04	07
Hadiur Rahman Hamim	2030550	04	07
Md Shakil Ahmed	2120290	04	07
Maruf Hasan Rion	2010597	04	07
Md Mahabub Alam Mahin	2030526	04	07
Hummah khadija Nirjana	2021868	01	01

Contents

CHAPTER 1 -INTRODUCTION:

A. BACKGROUND OF THE ORGANIZATION - IUB:

The Independent University, Bangladesh (IUB) has a number of strong and adaptable institutions, including the following:

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

The university has been an active participant in the growth of the education sector in Bangladesh and produced capable and knowledgeable scholars contributing both here and abroad.

IUB achieved this by working closely with the Ministry of Education, the University Grants Commission (UGC), and other necessary institutions for each of the schools, regularly updating the curriculum, implementing a system to track student performance based on a quantified approach between course curricula and standards set by UGC and the Bangladesh government, and continuously monitoring student performance.

The goal of this report is to review the student performance monitoring system that IUB currently employs, perform the necessary process analysis, and suggest a new and better system that reduces error, makes data analysis and report generation easier for all interested parties, and improves overall system performance produce/display important data that IUB and its partners will need to enhance academia and turn out better scholars. The first section explains the organization of the issue and the project we undertook for it in detail.

Our research into the current student performance monitoring system revealed many areas where beneficial changes could be made to speed up each step of the process, facilitate communication between important parties, reduce opportunities for mistakes and data duplication, and, most importantly, make it simpler for all stakeholders to sort through large datasets and find information that is relevant to their needs.

As we read through this report, we will go deeper into the inner workings of the current student performance monitoring system, the business processes entailed, the data management concerns and issues, and how we can create a better system to address these concerns and address them for fixing and improvement.

B. BACKGROUND OF THE PROJECT - SPMS 4.0:

Student Performance Monitoring System (SPMS 4.0) is an Outcome-Based Education (OBE) framework designed to assess the performance of students, instructors, schools, departments, and programs. It provides a comprehensive monitoring system that enables higher education authorities to identify areas for improvement and develop effective strategies accordingly. With SPMS 4.0, IUB can track student progress and evaluate the effectiveness of their curriculum, faculty, and overall performance, thus ensuring the continuous improvement of their educational programs.

C. OBJECTIVE OF THE PROJECT - SPMS 4.0:

SPMS 4.0 is an effective monitoring and analysis system that evaluates the performance of various stakeholders within an educational institution, including students, course instructors, departments, schools, and programs. This is achieved by utilizing a database of assessments, such as quizzes, midterm exams, and final term exams, to gather necessary data and documents, including exam question papers, answer scripts, course outlines, and marks achieved by students.

With this data, SPMS 4.0 assesses stakeholder performance based on their Course Outcomes (CO), Program Learning Outcomes (PLO), and Program Outcomes (PO), enabling students to monitor their own performance through statistical analysis. Additionally, Higher Authorities can take advantage of a wide range of analytical reports provided by SPMS 4.0 to draw valuable conclusions and make necessary improvements to the educational programs, departments, schools, and programs. This ensures continuous growth and development within the institution, enhancing the performance of all stakeholders involved.

D. SCOPE OF THE PROJECT:

After conducting a thorough analysis of the existing SPMS 3.0 system, we have identified several areas where the system is lacking. To address these shortcomings, we have redesigned the system to incorporate additional functionalities. The new system can accept, and store data related to Student ID, Educational year, semester, course, Enrolled section, and Obtained grade.

Students can now enter their data manually or upload a CSV file. If a CSV file is uploaded, the system will automatically extract the relevant data and insert it into the database. This feature greatly streamlines the data entry process, saving time and effort for both students and administrators.

In addition to these enhancements, students can now view their individual course outcomes (CO) in graphical form, based on their results. This feature provides students with a clear and easy-to-understand representation of their progress in each course, allowing them to identify areas for improvement and track their academic achievements more effectively.

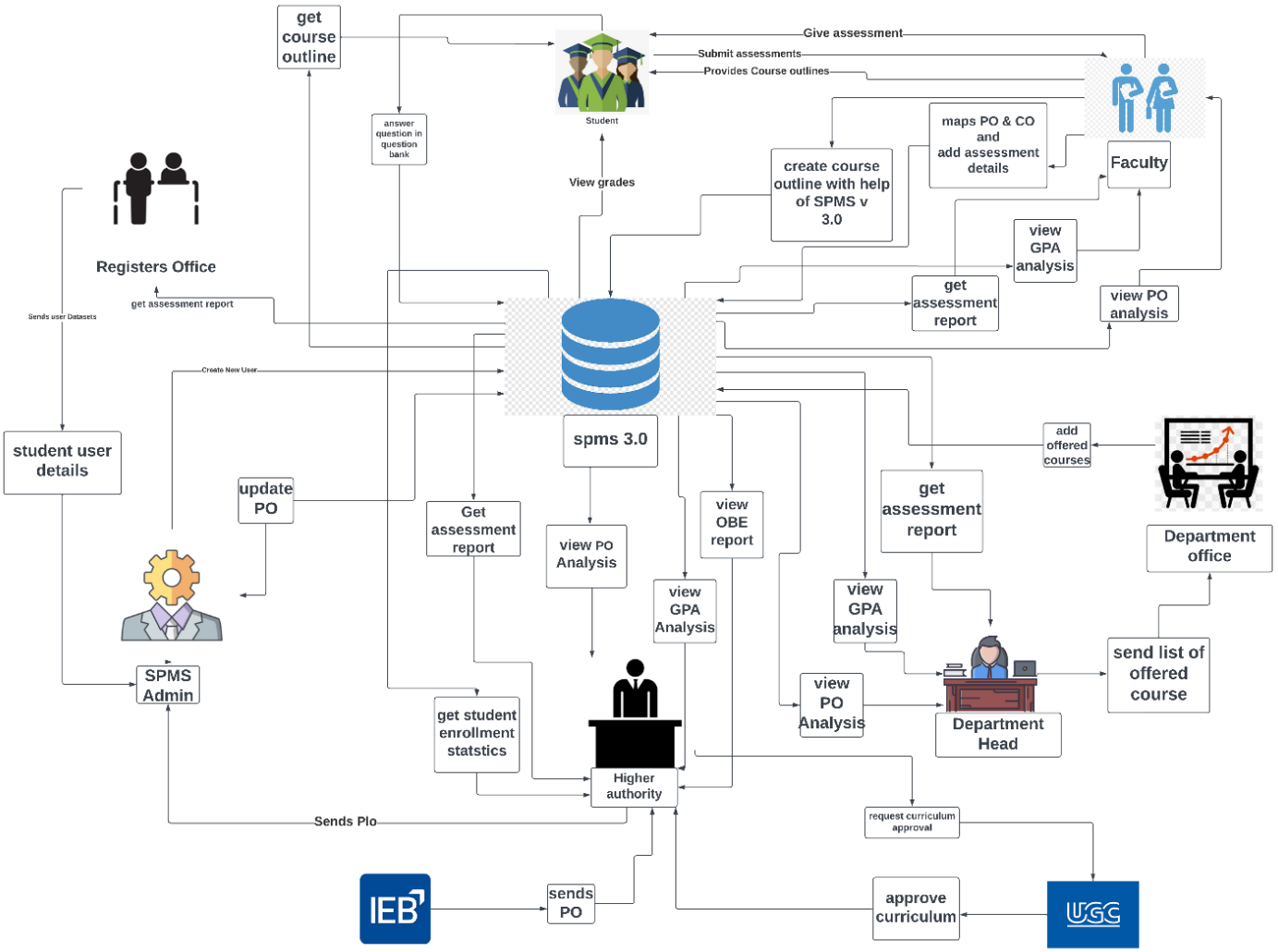
Overall, the revamped SPMS 3.0 system provides an improved user experience and greater functionality, making it easier and more efficient for students to manage their academic records and track their progress towards their educational goals.

CHAPTER-2 REQUIREMENT ANALYSIS

A. RICH PICTURE – EXISTING SYSTEM (SPMS 3.0):

Using a rich picture makes it simpler for everyone to comprehend how a system's processes work. It is composed of images, text, symbols, and icons that are all utilized to visually represent the scenario. [3] We can see connections and interconnections that we might otherwise miss if the picture is rich [3]. It aids

in identifying one or more themes that participants might want to investigate further and address. Rich images are thus always utilized in the pre-analysis stage [3].



In this rich picture the stakeholders are:

1) UGC

2) IEB

3) Higher Authority (VC, Dean etc)

4) Department Head

5) Department Office

6) SPMSV2.0 Admin (SPMS Manager)

7) Registers Office

8) Faculty

9) Student

The Main Storages are

1) SPMS V3.0

B.SIX ELEMENT ANALYSIS – EXISTING SYSTEM (SPMS 3.0)

From the rich picture we can see that there are 10 key processes:

1) Creating storing and giving Course Outline

2) Add Questions to the question bank and grading the answer script

3) Course based student performance trend according to GPA

4) Faculty based student performance according to GPA

5) Course wise PLO achievement of a student

6) Student performance trend under VC/Dean/Head of Department

7) Course, Program, department, school CLO-PLO statistics

8) Course, student, department school wise expected vs achieved PLO

9) Department average of total PLO achieved and attempted students

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

Head-wise.

We can use six element analysis to analyze the impact of six elements in a process here the six elements are

1. Human

2. Non computing Hardware

3. Computing Hardware

4. Software.

5. Database.

6. Network and Communication.

Process	Human	Non computing Hardware	Computing Hardware	Software	Database	Network and Commu-nication
Preparing storing and giving Course Outline	**Faculty:** 1) Signs into System using their ID and Password. 2) Select Create Course Outline Tab. 3) Select From the options that they wish to add in their course outline. 4) Press the Create button. 5) Store course outline into system. **Students:** 1) Signs into System using their ID and Password. 2) Select Course 3) View/Download Course Outline from System.		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer** 1) Used to print hard copy of course outlines if required.	**SPMS 3.0** 1)Used to store Data into the database	**SPMS 3.0** Data base 1) All valid data are stored here which can be updated by SPMS 3.0 admins.	1)Used to Sign into SPMS3.0
Add Questions to the question bank and grading the answer script	**Faculty:** 1) Signs into System using their ID and Password. 2) Select course and choose section's that has to solve the question. 3) Input the question in the question bank. 4) Press the Assign Button. 5) Grade the answers submitted by the students **Student:** 1) Signs into System using their ID and Password. 2) Answer the question assigned		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printe**r 1) Used to print the grades gotten by the whole section	**SPMS3.0** 1) Used to store Data into the database or generate result graph using data from the database.	**SPMS 3.0** Database 1) All valid data are stored here which can be updated by SPMS 3.0 admins.	**Internet** 1) Used to Sign into SPMS3.0

	by the faculty in the answer bank 3) Press the Submit button 4) Check grade in SPMS4.0 after faculty is done checking					
Course based student performance trend according to GPA	**Department Head:** 1) Signs into System using their ID and Password. 2) Input the time period and course ID to be viewed. 3) View student progress through a graph made after analysis and the GPA earned by maximum/minimum /Average students. **Faculty:** 1) Signs into system using their ID and Password. 2) Search for the course that they are teaching using course ID and time period and view the progress of that students of that course. **Student:** 1) Signs into System using their ID and Password. 2) Search for the course using course ID and View their progress of that course and the GPA they earned. **Dean/VC:** 1) Signs into system using their ID and Password.		**Computer/ Laptop** 1) Used to Sign into SPMS3.0. **Printer** 1) Used to print hard copy of the progress of current semester's students and compare with the progress of the previous semester's students who did that course.	**SPMS3. 0** 1)Used to store student Data into the database or generate perform -- once analysis graph using data from the database.	**SPMS 3.0 Database** 1) All valid data are stored here which can be updated by SPMS 3.0 admins.	**Internet** 1) Used to Sign into SPMS3.0

	2) Search for the course using course ID and time period and View the progress of the students of that course					
Faculty based student performance according to GPA	**Faculty:** 1) Signs into system using their ID and Password. 2) View the Progress of the students who are being taught by them. **Department Head:** 1) Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty's name. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students. **Dean/VC**: 1) Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty's name and Department ID. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students.		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer** 1) Used to print hard copy of the progress of students taught by a faculty	**SPMS3. 0** 1) Used to store student Data into the database or generate perform - ance analysis graph using data from the database.	**SPMS 3.0 Database** 1) All valid data are stored here which can be updated by SPMS 3.0 admins.	**Internet** 1) Used to Sign into SPMS3.0

Course wise PLO achievement of a student	**VC/ Dean:** 1) Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the student. **Department Head:** 1) Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the students. **Faculty:** 1) Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the students in a course. **Student:** 1) Signs into system using their ID and Password. 2) View PLOs they have achieved so far and how many they need to achieve to complete the course.		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer** 1) Used to print hard copy of a report of students who completed most the PLO achievements If needed.	**SPMS3. 0** 1) Used to store Data and generate PLO automati-cally based on the CO provide d.	**SPMS 3.0 Database** 1) All valid data are stored here which can be updated by SPMS 3.0 admins.	Internet 1) Used to Sign into SPMS3.0

Student performance trend Under VC/Dean/ Head of Department	**Dean:** 1) Signs into system using their ID and Password. 2) Search for Department Head to be checked using their Name and Department ID. 3) View student progress under them or them. **VC:** 1) Signs into system using their ID and Password. 2) Search for a Dean or Department Head to be checked using their Name and either School ID or Department ID. 3) View student progress under them. **Department Head:** 1) Signs into system using their ID and Password. 2) View student progress under them		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer** 1) Used to print the hard copy of the progress report if needed	**SPMS3. 0** 1) Used to store Data into the database or generate perform ance analysis graph using data from the database.	**SPMS 2.0** Database 1) All valid data are stored here which can be updated by SPMS 3.0 admins.	**Internet** 1) Used to Sign into SPMS3.0

Course, Program, department, school CLO-PLO statistics	**Dean/VC:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students. **Department Head**: 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students. **Faculty:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students. **Student:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by them and other students.		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer** 1) Used to print the hard copy of the progress report if needed	**SPMS3. 0** 1) Used to store Data into the database and generate CLO-PLO statistic al data or graphs.	**SPMS 3.0 Database** 1) All valid data are stored here which can be updated by SPMS 3.0 admins.	**Internet** 1) Used to Sign into SPMS3.0
Course, student, department school wise expected vs achieved PLO	**Dean/VC:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer**	**SPMS3. 0** 1) Used to store Data into the database or generate perform ance analysis	**SPMS 3.0 Database** 1) All valid data are stored here which can be updated by SPMS	**Internet** 1) Used to Sign into SPMS3.0

	between expected and achieved. **Department Head:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved. **Faculty:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved. **Student:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved.		1) Used to print the hard copy of both the previous and current semester's achieved PLO to compare.	graph using data from the database	3.0 admins	
Department average of total PLO achieved and attempted students	**Dean/VC:** 1) Sign into the system using ID and Password. 2) Enter the time period of the semester wished to be viewed. 3) View the departmental average of total PLO achieved along with the number of students who attempted.		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer** 1) Used to print the hard copy of PLO reports	**SPMS3. 0** 1) Used to store Data into the database or generate perform ance analysis graph using data from the database.	**SPMS 3.0 Database** 1) All valid data are stored here which can be updated by SPMS 3.0 admins	**Internet** 1) Used to Sign into SPMS3.0

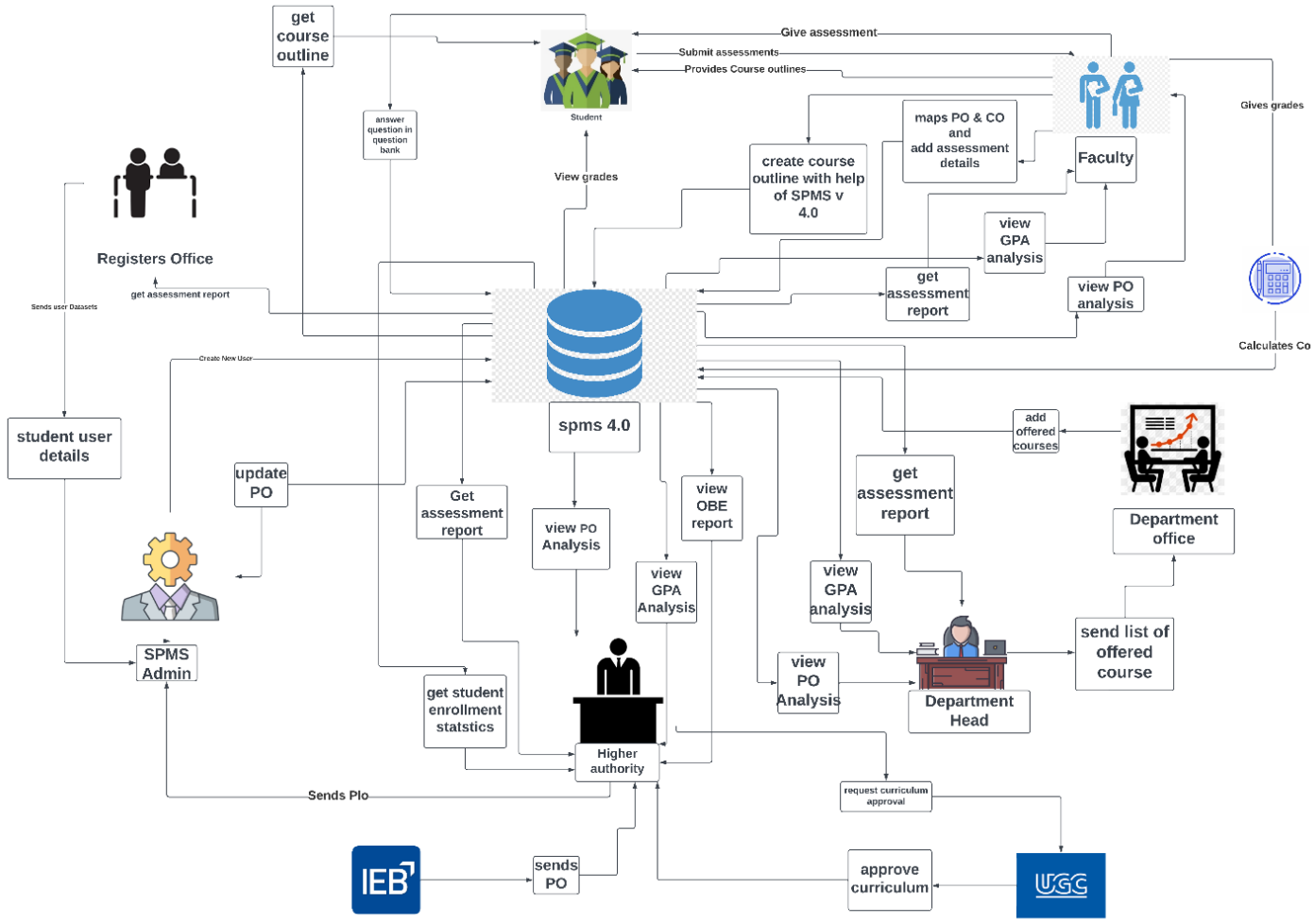
	Department Head: 1) Sign into the system using ID and Password. 2) Enter the time period of the semester wished to be viewed. 3) View the departmental average of total PLO achieved along with the number of students who attempted. **Faculty:** 1) Sign into the system using ID and Password. 2) View the total departmental average of the PLO achieved by the students. **Student:** 1) Sign into the system using ID and Password. 2) View the total departmental average of the PLO achieved by the students					
Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.	**VC** 1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester. **Dean** 1) Sign into the system using ID and Password.		**Computer/ Laptop** 1) Used to Sign into SPMS3.0 **Printer** 1) Used to print the hard copy of Student Enrollment Statistics If Needed.	**SPMS3. 0** 1) Used to store Data into the database and generate Student Enrollment Statistics graphs	**SPMS 3.0 Database** 1) All valid data are stored here which can be updated by SPMS 3.0 admins	**Internet** 1) Used to Sign into SPMS3.0

	2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester. **Department Head** 1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester					

C. PROBLEM ANALYSIS – EXISTING SYSTEM (SPMS 3.0):

Process Name	Stake Holders	Concerns (Problems)	Analysis (Reason of the Problem)	Proposed Solution
Assessments and Grading	Faculty and Student	CLO calculation for any student was done manually.	No implemented system was present to calculate and show the CLO in graph or any kind of chart.	Building a system which can take the required data and return a calculated CLO of the given grade
Data entry in a bulk	Faculty and Student	Entry for the calculation of the CLO cannot be given in a bulk. Like the csv file can't be uploaded wholly.	There is no such system where the csv file can be uploaded and processed.	An option in the CLO calculation system should be kept where the csv file can be uploaded.
Storing backup data	Faculty	1)There is no backup data system where it can be seen that which data was inserted by which faculty. 2)There is no time column where the time of data entry should be recorded during the entry.	No such table was created for the backup of the entry record.	A new table should be created where all the records are present which can be viewed by the admin in case of any requirement.

D. RICH PICTURE – PROPOSED SYSTEM (SPMS 4.0)



In this rich picture the stakeholders are:

1) UGC

2) IEB

3) Higher Authority (VC, Dean etc)

4) Department Head

5) Department Office

6) SPMSV2.0 Admin (SPMS Manager)

7) Registers Office

8) Faculty

9) Student

The Main Storage is

1) SPMS V4.0

E. SIX ELEMENT ANALYSIS – PROPOSED SYSTEM (SPMS 4.0):

From the rich picture we can see that there are 9 key processes:

1) Creating storing and giving Course Outline

2) Add Questions to the question bank and grading the answer script

3) Course based student performance trend according to GPA

4) Faculty based student performance according to GPA

5) Course wise PLO achievement of a student

6) Student performance trend under VC/Dean/Head of Department

7) Course, Program, department, school CLO-PLO statistics

8) Course, student, department school wise expected vs achieved PLO

9) Department average of total PLO achieved and attempted students

10) Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.

We can use six element analysis to analyze the impact of six elements in a process here the six elements are.

1. Human

2. Non computing Hardware

3. Computing Hardware

4. Software.

5. Database.

6. Network and Communication.

Process	Human	Non computing Hardware	Computing Hardware	Software	Database	Network and Commu-nication
Preparing storing and giving Course Outline	**Faculty:** 1) Signs into System using their ID and Password. 2) Select Create Course Outline Tab. 3) Select From the options that they wish to add in their course outline. 4) Press the Create button. 5) Store course outline into system. **Students:** 1) Signs into System using their ID and Password. 2) Select Course 3) View/Download Course Outline from System.		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer** 1) Used to print hard copy of course outlines if required.	**SPMS 4.0** 1)Used to store Data into the database	**SPMS 4.0** Data base 1) All valid data are stored here which can be updated by SPMS 4.0 admins.	1)Used to Sign into SPMS 4.0
Add Questions to the question bank and grading the answer script	**Faculty:** 1) Signs into System using their ID and Password. 2) Select course and choose section's that has to solve the question. 3) Input the question in the question bank. 4) Press the Assign Button. 5) Grade the answers submitted by the students **Student:** 1) Signs into System using their ID and Password. 2) Answer the question assigned by the faculty in the answer bank		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printe**r 1) Used to print the grades gotten by the whole section	**SPMS 4.0** 1) Used to store Data into the database or generate result graph using data from the database.	**SPMS 4.0** Database 1) All valid data are stored here which can be updated by SPMS 4.0 admins.	**Internet** 1) Used to Sign into SPMS 4.0

	3) Press the Submit button 4) Check grade in SPMS4.0 after faculty is done checking					
Course based student performance trend according to GPA	**Department Head:** 1) Signs into System using their ID and Password. 2) Input the time period and course ID to be viewed. 3) View student progress through a graph made after analysis and the GPA earned by maximum/minimum /Average students. **Faculty:** 1) Signs into system using their ID and Password. 2) Search for the course that they are teaching using course ID and time period and view the progress of that students of that course. **Student:** 1) Signs into System using their ID and Password. 2) Search for the course using course ID and View their progress of that course and the GPA they earned. **Dean/VC:** 1) Signs into system using their ID and Password.		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0. **Printer** 1) Used to print hard copy of the progress of current semester's students and compare with the progress of the previous semester's students who did that course.	**SPMS 4. 0** 1)Used to store student Data into the database or generate perform -- once analysis graph using data from the database.	**SPMS 4.0 Database** 1) All valid data are stored here which can be updated by SPMS 4.0 admins.	**Internet** 1) Used to Sign into SPMS 4.0

	2) Search for the course using course ID and time period and View the progress of the students of that course					
Faculty based student performance according to GPA	**Faculty:** 1) Signs into system using their ID and Password. 2) View the Progress of the students who are being taught by them. **Department Head:** 1) Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty's name. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students. **Dean/VC**: 1) Signs into system using their ID and Password. 2) Search for a faculty to be assessed using the faculty's name and Department ID. 3) View the Progress of the students who are being taught under that faculty basing on the GPA earned by the students.		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer** 1) Used to print hard copy of the progress of students taught by a faculty	**SPMS 4. 0** 1) Used to store student Data into the database or generate perform - ance analysis graph using data from the database.	**SPMS 4.0 Database** 1) All valid data are stored here which can be updated by SPMS 4.0 admins.	**Internet** 1) Used to Sign into SPMS 4.0

Course wise PLO achievement of a student	**VC/ Dean:** 1) Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the student. **Department Head:** 1) Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the students. **Faculty:** 1) Signs into system using their ID and Password. 2) Select PLO achievement Tab and search using Course ID 3) View PLOs achieved by the students in a course. **Student:** 1) Signs into system using their ID and Password. 2) View PLOs they have achieved so far and how many they need to achieve to complete the course.		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer** 1) Used to print hard copy of a report of students who completed most the PLO achievements If needed.	**SPMS 4. 0** 1) Used to store Data and generate PLO automati-cally based on the CO provide d.	**SPMS 4.0 Database** 1) All valid data are stored here which can be updated by SPMS 4.0 admins.	Internet 1) Used to Sign into SPMS 4.0

Student performance trend Under VC/Dean/ Head of Department	**Dean:** 1) Signs into system using their ID and Password. 2) Search for Department Head to be checked using their Name and Department ID. 3) View student progress under them or them. **VC:** 1) Signs into system using their ID and Password. 2) Search for a Dean or Department Head to be checked using their Name and either School ID or Department ID. 3) View student progress under them. **Department Head:** 1) Signs into system using their ID and Password. 2) View student progress under them		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer** 1) Used to print the hard copy of the progress report if needed	**SPMS 4. 0** 1) Used to store Data into the database or generate perform ance analysis graph using data from the database.	**SPMS 4.0** Database 1) All valid data are stored here which can be updated by SPMS 4.0 admins.	**Internet** 1) Used to Sign into SPMS 4.0

Course, Program, department, school CLO-PLO statistics	**Dean/VC:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students. **Department Head**: 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students. **Faculty:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by students. **Student:** 1) Signs into system using their ID and Password. 2) View CLO-PLO mapped statistics achieved by them and other students.		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer** 1) Used to print the hard copy of the progress report if needed	**SPMS 4. 0** 1) Used to store Data into the database and generate CLO-PLO statistic al data or graphs.	**SPMS 4.0 Database** 1) All valid data are stored here which can be updated by SPMS 4.0 admins.	**Internet** 1) Used to Sign into SPMS 4.0
Course, student, department school wise expected vs achieved PLO	**Dean/VC:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer**	**SPMS 4. 0** 1) Used to store Data into the database or generate perform ance analysis	**SPMS 4.0 Database** 1) All valid data are stored here which can be updated by SPMS	**Internet** 1) Used to Sign into SPMS 4.0

	between expected and achieved. **Department Head:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved. **Faculty:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved. **Student:** 1) Sign into the system using ID and Password. 2) View the achieved PLO of the students during time entered that has been inputted and comparison between expected and achieved.		1) Used to print the hard copy of both the previous and current semester's achieved PLO to compare.	graph using data from the database	4.0 admins	
Department average of total PLO achieved and attempted students	**Dean/VC:** 1) Sign into the system using ID and Password. 2) Enter the time period of the semester wished to be viewed. 3) View the departmental average of total PLO achieved along with the number of students who attempted.		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer** 1) Used to print the hard copy of PLO reports	**SPMS 4. 0** 1) Used to store Data into the database or generate perform ance analysis graph using data from the database.	**SPMS 4.0 Database** 1) All valid data are stored here which can be updated by SPMS 4.0 admins	**Internet** 1) Used to Sign into SPMS 4.0

	Department Head: 1) Sign into the system using ID and Password. 2) Enter the time period of the semester wished to be viewed. 3) View the departmental average of total PLO achieved along with the number of students who attempted. **Faculty:** 1) Sign into the system using ID and Password. 2) View the total departmental average of the PLO achieved by the students. **Student:** 1) Sign into the system using ID and Password. 2) View the total departmental average of the PLO achieved by the students					
Student Enrollment Statistics VC-wise, Dean-wise, Department Head-wise.	**VC** 1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester. **Dean** 1) Sign into the system using ID and Password.		**Computer/ Laptop** 1) Used to Sign into SPMS 4.0 **Printer** 1) Used to print the hard copy of Student Enrollment Statistics If Needed.	**SPMS 4. 0** 1) Used to store Data into the database and generate Student Enrollment Statistics graphs	**SPMS 4.0 Database** 1) All valid data are stored here which can be updated by SPMS 4.0 admins	**Internet** 1) Used to Sign into SPMS 4.0

	2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester. **Department Head** 1) Sign into the system using ID and Password. 2) Select Student Enrollment Statistics tab and select Year and Semester under that tab 3) View Student Enrollment Statistics of That Year and Semester					
	Student: Enters 1) ID 2) Course ID 3) Section 4) Semester 5) Year		**Computer/ Laptop** Used to Sign into SPMS 4.0 **Printer** Used to print the hard copy of Student Enrollment Statistics If Needed.	**SPMS 4.0** Used to store Data into the database and generate Student Enrollment Statistics graphs	**SPMS 4.0 Database** All valid data are stored here which can be updated by SPMS 4.0 admins	**Internet** Used to Sign into SPMS 4.0

CHAPTER 3 - LOGICAL SYSTEM DESIGN

We will go through the steps of building a data model for our suggested system in this chapter so that the data can be stored in a database. This data model is a conceptual representation of the relationships between various data objects, the rules, and the data objects themselves. Data modeling facilitates the visual representation of data and ensures that it complies with legal requirements, business regulations, and governmental directives. Data models guarantee data quality while guaranteeing uniformity in naming standards, default values, semantics, and security. Our proposed system will be created to better represent all the data.

A. BUSINESS RULES – SPMS 4.0:

1. A Student must have one department. A STUDENT has StudentID, FirstName, LastName, dateOfBirth, gender, email, phone, address, departmentID, programID, enrollmentYear, enrollmentSemester, password. A department must have one or many Students.

2. A Student may perform many registrations. A REGISTRATION includes RegistrationID, sectionID, studentID. A Registration must be performed by at least one student.

3. A section mandatorily has many registrations. A registration has at least one section. A section includes sectionID, sectionNum, courseID, facultyID, year.

4. A registration may belong to many EVALUATIONS. An evaluation mandatorily belongs to one registration. An EVALUATION contains evaluationID, examID, registrationID, totalMarks.

5. A CO must map with one PO. A PO's must map with one or many CO's. PO includes poID, poNum, programID.

6. A PO must contain one program. A program contains one or many PO's. A PROGRAM has programID, programName, departmentID. A program must contain one or many courses. A Course must contain one course.

7. A program must belong to one department. A department must belong to one or many programs. A DEPARTMENT contains departmentID, departmentName, schoolID.

8. A department must contain one school. A SCHOOL must contain one or many departments. A school includes schoolID, schoolName.

9. An employee has four sub-types (Dean, Department Head, Faculty). An EMPLOYEE includes employeeID, password, firstName, lastName.

10. A school must be run by exactly one. A dean must run exactly one school. A DEAN has schoolID, startDate, endDate.

11. A Department must be run by exactly one Department head. A department head must manage exactly one department. A DEPARTMENTHEAD includes departmentID, startDate, endDate.

12. A Faculty must have exactly one Department. A department must have one or many Faculties. A FACULTY includes departmentID, rank, joinDate. A faculty may teach many sections. A section must be taught by exactly one faculty.

13. 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.

14. 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.

15. 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, learningStrategy, assessmentStrategy, correspondingClo, courseOutlineID.

16. An EXAM has exactly one EVALUATION. An EVALUATION for an EXAM is done exactly once. An exam belongs to exactly one section. An EXAM includes examID, examName, sectionID. A section must have one or many exams.

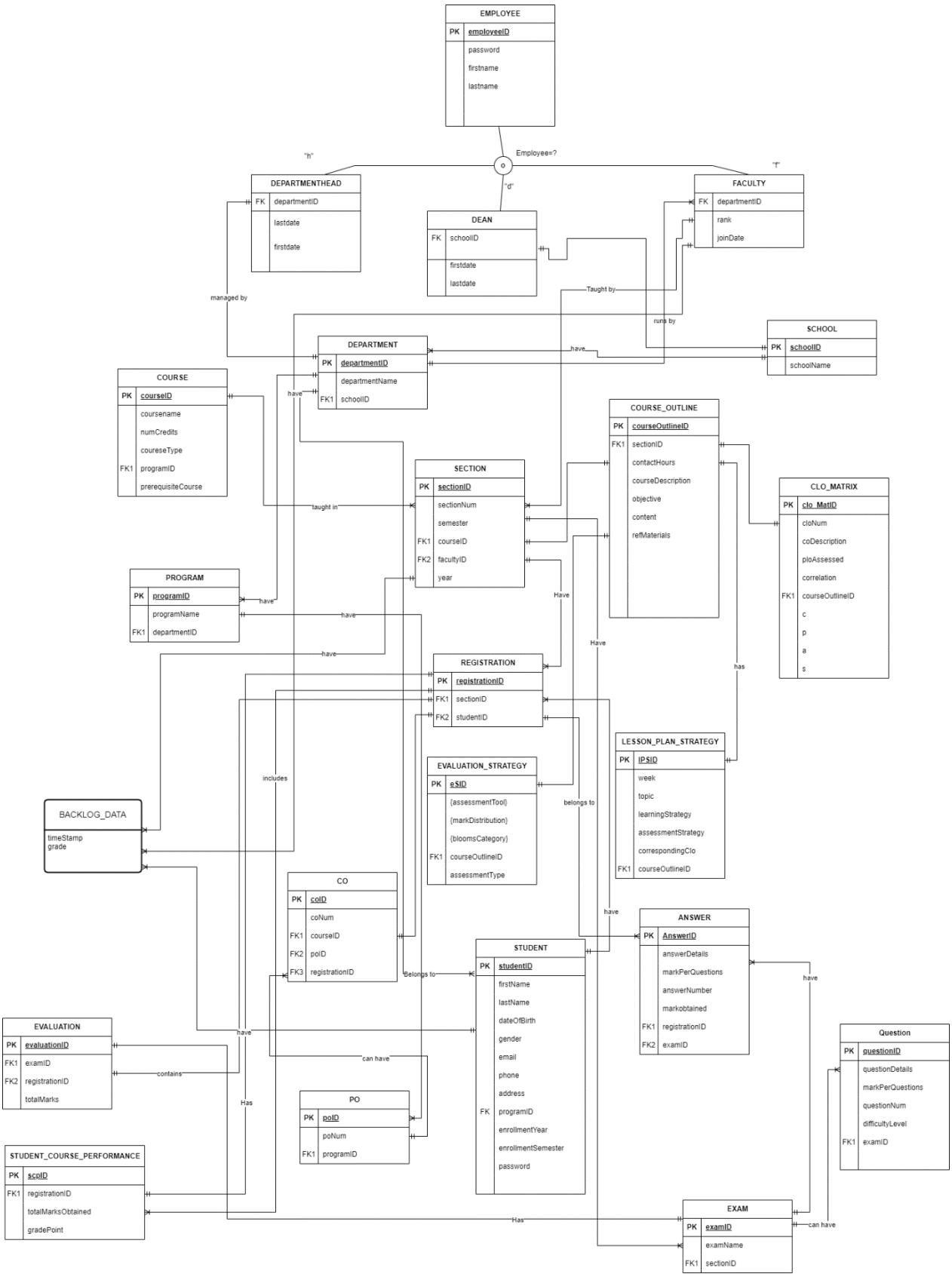
17. An EXAM must have one or many questions. Every QUESTION must belong to exactly one exam. A QUESTION includes questionID, questionDetails, marksPerQuestion, questionNum, difficutltyLevel, examID, coNum. A Question is answered exactly once. An ANSWER has exactly one question.

18. A Section belongs to exactly one Course. A course must have one or many Section. Course includes courseID, courseName, numCredits, courseType, programID, prerequisiteCourse.

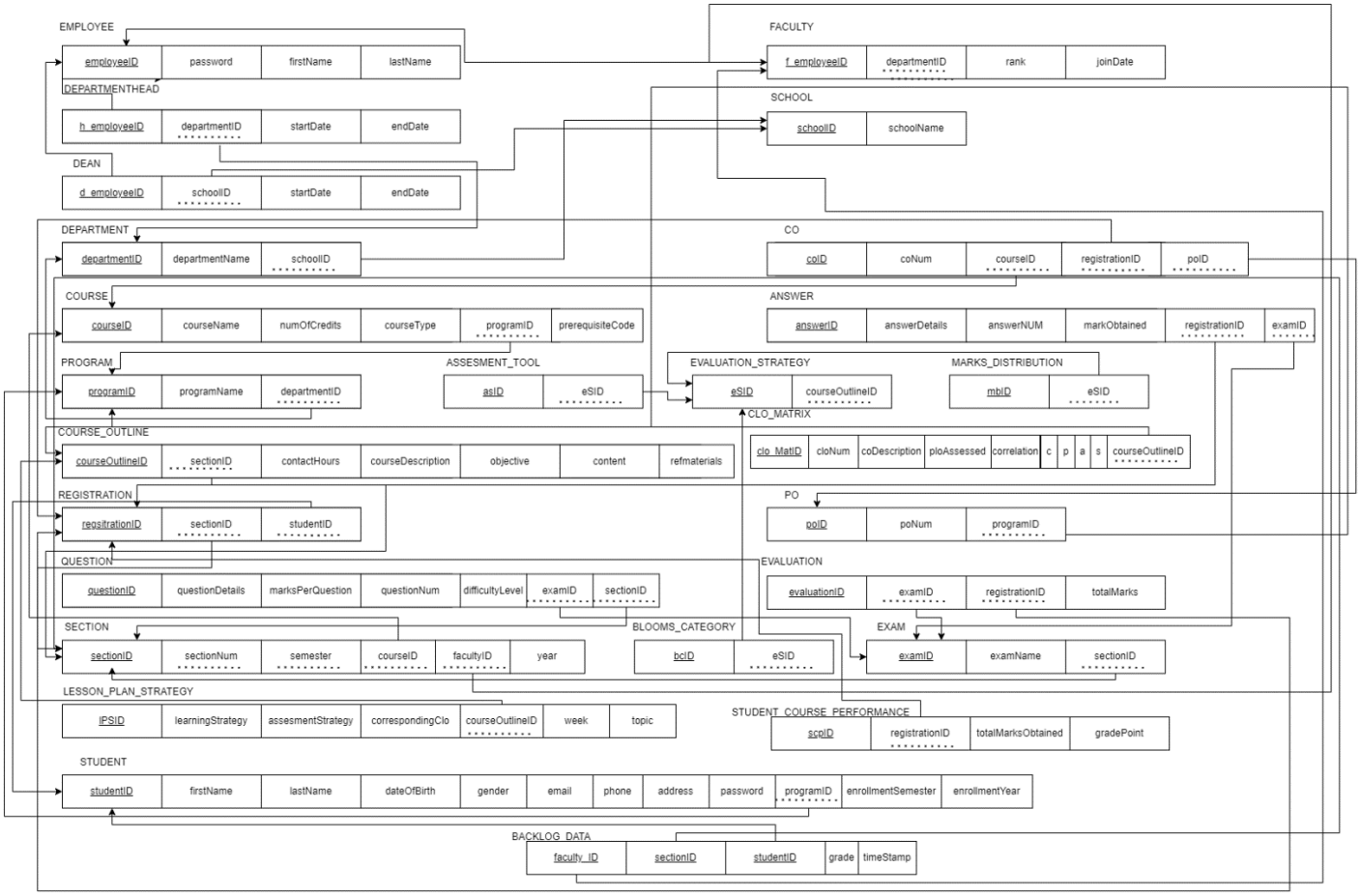
19. A Student Course Performance evaluation is done for Registration exactly once. A REGISTRATION has student course performance evaluation done exactly once. A REGISTRATION has exactly one evaluation. An Evaluation has exactly one Registration.

20. A Student must be enrolled in one or many Section, A section will have many students, A Student must be assigned many CO and CO is assigned to many Students , A Section must be assigned one or many CO and CO is must assigned of one or many Section and here CLO_CAL is recorded in the database.

B. ENTITY RELATIONSHIP DIAGRAM:



C. ENTITY RELATIONSHIP DIAGRAM TO RELATIONAL SCHEMA:



D. NORMALIZATION:

EMPLOYEE (e)	employeeID	E1
	password	E2
	firstName	E3
	lastName	E4
DEAN (d)	d_employeeID	D1
	startDate	D2
	endDate	D3
	schoolID	A1
DEPARTMENT_HEAD (h)	h_employeeID	H1
	departmentID	X1
	startDate	H2
	endDate	H3
FACULTY (f)	f_employeeID	F1
	departmentID	X1
	rank	F2
	joinDate	F3
SCHOOL (a)	schoolID	A1
	schoolName	A2
DEPARTMENT (x)	departmentID	X1
	departmentName	X2
	schoolID	A1
PROGRAM (p)	programID	P1
	programName	P2
	departmentID	X1
COURSE (c)	courseID	C1
	courseName	C2
	numOfCredits	C3
	courseType	C4
	programID	P1
	prerequisitecode	C5
SECTION (b)	sectionID	B1
	sectionNum	B2
	semester	B3
	courseID	C1
	f_employeeID	F1
	year	B4
STUDENT (s)	studentID	S1
	firstName	S2
	lastName	S3
	dateOfBirth	S4
	gender	S5
	address	S6
	email	S7

	phone	S8
	programID	P1
	enrollmentSemester	S9
	enrollmentYear	S10
	password	S11
COURSE_OUTLINE (o)	courseOutlineID	O1
	sectionID	B1
	contactHours	O2
	courseDescription	O3
	objective	O4
	content	O5
	refMaterials	O6
REGISTRATION (r)	registrationID	R1
	sectionID	B1
	studentID	S1
PO (v)	poID	V1
	poNum	V2
	programID	P1
CO (t)	coID	T1
	coNum	T2
	poID	V1
	courseID	C1
	registrationID	R1
BACKLOG_DATA (m)	facultyID	F1
	studentID	S1
	sectionID	B1
	grade	M1
	timeStamp	M2
QUESTION (q)	questionID	Q1
	questionDetails	Q2
	marksOerQuestion	Q3
	questionNum	Q4
	difficultyLevel	Q5
	examID	Y1
	courseID	C1
CLO_MATRIX (n)	Clo_MatID	N1
	cloNum	N2
	coDescription	N3
	ploAssessed	N4
	correlation	N5
	c	N6
	p	N7
	a	N8
	s	N9
	courseOutlineID	O1
ANSWER (w)	answerID	W1
	answerDetails	W2
	answerNum	W3

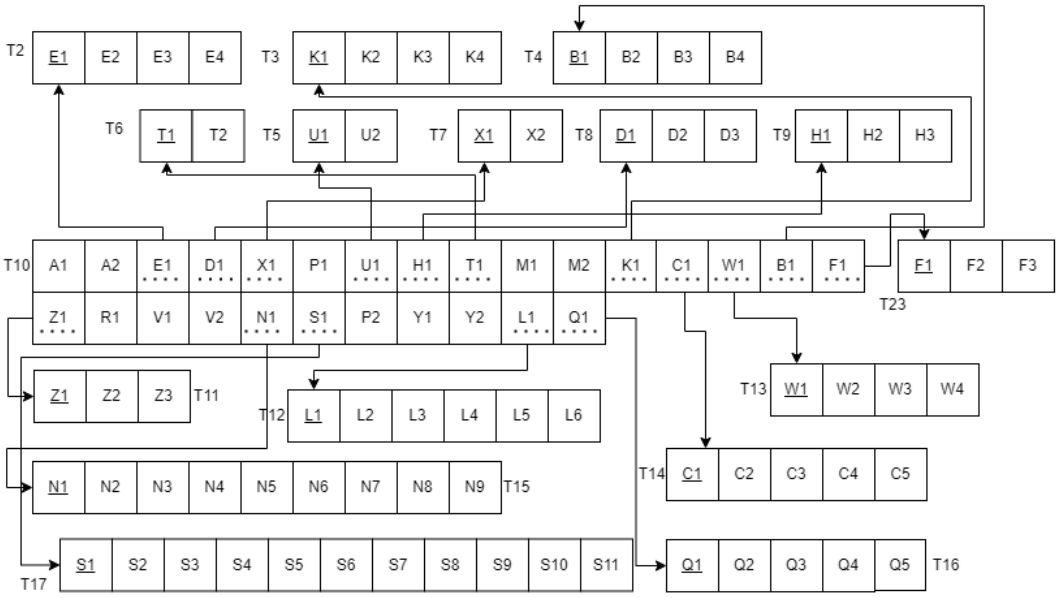
	marksObtained	W4
	registrationID	R1
	examID	Y1
EVALUATION_STRATEGY (k)	eSID	K1
	assesmentTool	K2
	marksDistribution	K3
	bloomsCategory	K4
	courseOutlineID	O1
EVALUATION (u)	evaluationID	U1
	examID	Y1
	registrationID	R1
	totalMarks	U2
STUDENT_COURSE_PERFORMANCE (z)	scpID	Z1
	registrationID	R1
	totalMarksObtained	Z2
	gradePoint	Z3
LESSON_PLAN_STRATEGY (l)	lPSID	L1
	week	L2
	topic	L3
	learningStrategy	L4
	assesmentStrategy	L5
	correspondingClo	L6
	courseOutlineID	O1
EXAM (y)	examID	Y1
	examName	Y2
	sectionID	B1

1NF:
There are no repeating groups and there is atleast one primary key .

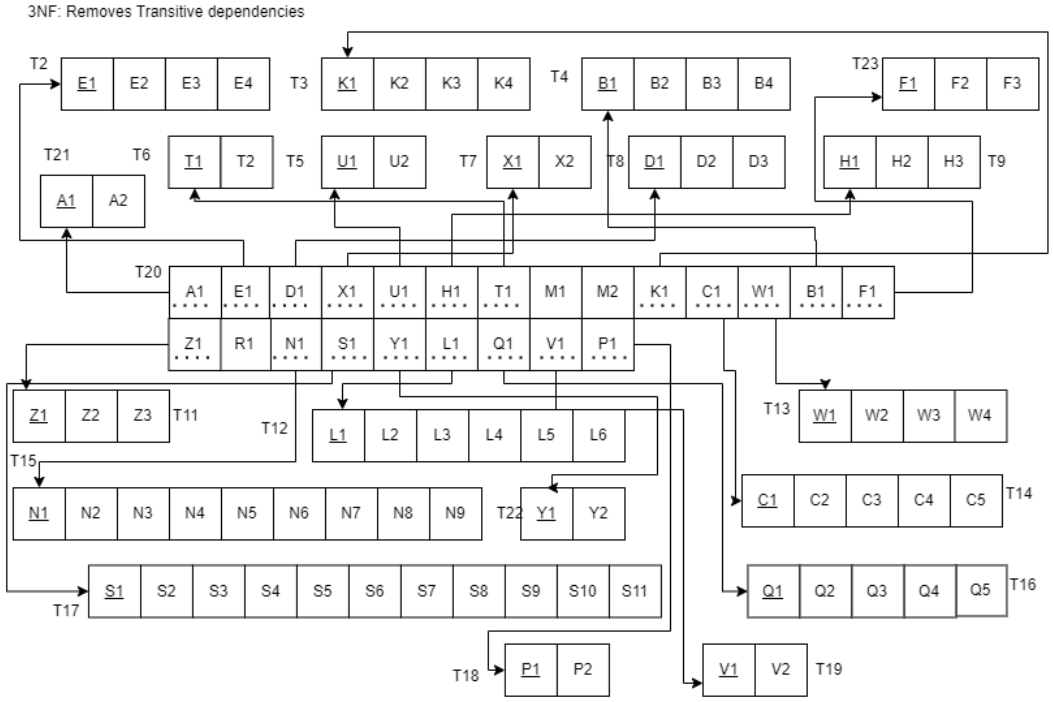
T1

A1	A2	E1	E2	E3	E4	D1	D2	D3	F1	F2	F3	H1	H2	H3	X1	X2	P1
P2	O1	O2	O3	O4	O5	O6	R1	V1	V2	N1	N2	N3	N4	N5	N6	N7	N8
N9	W1	W2	W3	W4	K1	K2	K3	K4	U1	U2	C1	C2	C3	C4	C5	B1	B2
B3	B4	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	T1	T2	M1	M2	
	Z1	Z2	Z3	L1	L2	L3	L4	L5	L6	Y1	Y2	Q1	Q2	Q3	Q4	Q5	

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



3NF: Removes Transitive dependencies



Already in BCNF

E.DATA DICTIONARY:

VC_T

Name		Data Type	Size	Remark
v_employeeID		INTEGER	11	This is the foreign key from the Employee table. E.g: "4250"
startDate		DATE		This is starting date for the VC. E.g: "01-03-2020"
endDate		DATE		This is the date VC retire from his post. E.g: "01-03-2024"

STUDENT_T

Name	Data Type	Size	Remark
studentID	INTEGER	11	This is the primary key for the Student table. E.g: "1921834".
firstName	VARCHAR	30	This is the first name of the student. E.g: "Rakibul".
lastName	VARCHAR	30	This is the last name of the student. E.g: "Hasan".

dateOfBirth	DATE		This is the birth date of the student. E.g: “2112-1996”.
gender	VARCHAR	6	This is the gender of the student. E.g: “Female”.
email	VARCHAR	30	This is the email of the student. E.g: “1921834@iub.edu.bd”
phone	VARCHAR	11	This is the phone of the student. E.g: “01XXXXXXXXX”.
address	VARCHAR	50	This is the address of the student. E.g: “House 1,Road 4,Block D, Bashundhara RA
departmentID	VARCHAR	3	This is the foreign key from the Department table. E.g: “CSE”
programID	INTEGER	11	This is the foreign key from the Program table. E.g: “1”
enrollmentSemester	VARCHAR	10	This is the enrollment semester of the student.
enrollmentYear	VARCHAR	4	This is enrollment year of the student.

STUDENT_COURSE_PERFORMANCE_T

Name	Data Type	Size	Remark
scpID	INTEGER	11	This is the primary key for this table
registrationID	INTEGER	11	This is the foreign key from registration table
totalMarksObtained	INTEGER	11	This is the total marks obtained by the student
gradePoint	FLOAT		This is the grade point achieved by the student

SECTION_T

Name	Data Type	Size	Remark
sectionID	INTEGER	11	This is the Primary Key for Section. E.g: “1”
sectionNum	INTEGER	11	This is the section number. E.g: “1”
semester	VARCHAR	6	This is the semester of the section. E.g: “Summer”
courseID	VARCHAR	6	This is the foreign key from the Course table. E.g: “CSE101”

facultyID	INTEGER	11	This is the foreign key from Faculty table. E.g: “1801”
year	YEAR	4	This is the year this section of this course was taken by this specific faculty

SCHOOL_T

Name	Data Type	Size	Remark
schoolID	VARCHAR	5	This is the primary key of School. E.g: “SETS”
schoolName	VARCHAR	50	This is the name of the School. E.g: “School of Engineering,
			Technology & Science”.

REGISTRATION_T

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

QUESTION_T

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

PROGRAM_T

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

PO_T

Name	Data Type	Size	Remark
poID	VARCHAR	5	This is the primary key for Program Outcome. E.g: “PO1”
poNum	INTEGER	11	This is the PO number. E.g: “1”
programID	INTEGER	11	This is a foreign key from Program table. E.g: “1”

PLO_T

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

LESSON_PLAN_STRATEGY_T

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

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

FACULTY_T

Name	Data Type	Size	Remark
f_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g: “4250”
departmentID	VARCHAR	3	This is the DepartmentID of the department faculty belongs to. E.g: “CSE”
rank	VARCHAR	30	This is the rank of the faculty. E.g: “Assistant Professor”
joinDate	DATE		This is starting date. E.g: “01-03-2020”

EXAM_T

Name	Data Type	Size	Remark

examID	INTEGER	11	This is the primary key for this table
examName	VARCHAR	30	This is the name of the exam
sectionID	INTEGER	11	This is the foreign key from exam table

EVALUATION_T

Name	Data Type	Size	Remark
evaluationID	INTEGER	11	This is the primary key for this table
examID	VARCHAR	20	This is the foreign key from exam table
registrationID	INTEGER	11	This is the foreign key from registration table
totalMarks	INTEGER	11	This is the total marks achieved by the student in a specific exam

EVALUATION_STRATEGY_T

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

ASSESSMENT_TOOL_T

Name	Data Type	Size	Remark

asID	INTEGER	11	This is the primary key for this table
eSID	INTEGER	11	This is the foreign key from evaluation strategy table

MARK_DISTRIBUTION_T

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

BLOOMS_CATEGORY_T

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

EMPLOYEE_T

Name	Data Type	Size	Remark
employeeID	INTEGER	11	This is the primary key for Employee table. E.g: “1801”
password	VARCHAR	10	This is the password of the employee

firstName	VARCHAR	50	This is the last name of the faculty. E.g: “Ahmed”
lastName	VARCHAR	50	This is the last name of the faculty. E.g: “Ahmed”

DEPARTMENTHEAD_T

Name	Data Type	Size	Remark
h_employeeID	INTEGER	11	This is the foreign key from the Employee table.
			E.g: “4250”
departmentID	VARCHAR	3	This is the DepartmentID of the department HEAD manages. E.g: “CSE”
startDate	DATE		This is starting date. E.g: “01-03-2020”
endDate	DATE		This is the date HEAD retire from his post. E.g: “0103-2024”

DEPARTMENT_T

Name	Data Type	Size	Remark
departmentID	VARCHAR	3	This is the primary key for the Department table. E.g: “CSE”

departmentName	VARCHAR	50	This is the name of the department. E.g: “Computer Science and Engineering”.
schoolID	VARCHAR	5	This is a foreign key from the School table. E.g: “SETS”.

DEAN_T

Name	Data Type	Size	Remark
d_employeeID	INTEGER	11	This is the foreign key from the Employee table. E.g: “4250”
schoolID	VARCHAR	5	This is the SchoolID of the school DEAN manages. E.g: “SETS”
startDate	DATE		This is starting date. E.g: “01-03-2020”
endDate	DATE		This is the date DEAN retire from his post. E.g: “0103-2024”

COURSE_T

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

COURSE_OUTLINE_T

Name	Data Type	Size	Remark
courseOutlineID	INTEGER	11	This is the primary key for this table

sectionID	INTEGER	11	This is the foreign key from the section table
courseDescription	MEDIUMTEXT		This is the description of the course
objective	MEDIUMTEXT		This is the objective of the course
content	MEDIUMTEXT		This is the content of the course
refMaterials	MEDIUMTEXT		This is the reference material
courseTitle	VARCHAR	1000	This is the title of the course
prerequsiteCode	VARCHAR	6	This is the prerequisite course code
creditValue	INTEGER	11	This is the credit value of the course

CO_T

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

poID	VARCHAR	6	This is the foreign key from the PLO table. E.g: “PO1”

CLO_MATRIX_T

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

ANSWER_T

Name	Data Type	Size	Remark
answerID	INTEGER	11	This is the primary key for this table

answerDetails	MEDIUMTEXT		This is the answer details
answerNum	INTEGER	11	This is the number of the answer
markObtained	INTEGER	11	This is the mark obtained by the student for each answer
registrationID	INTEGER	11	This is the foreign key from registration table
examID	INTEGER	11	This is the foreign key from the exam table

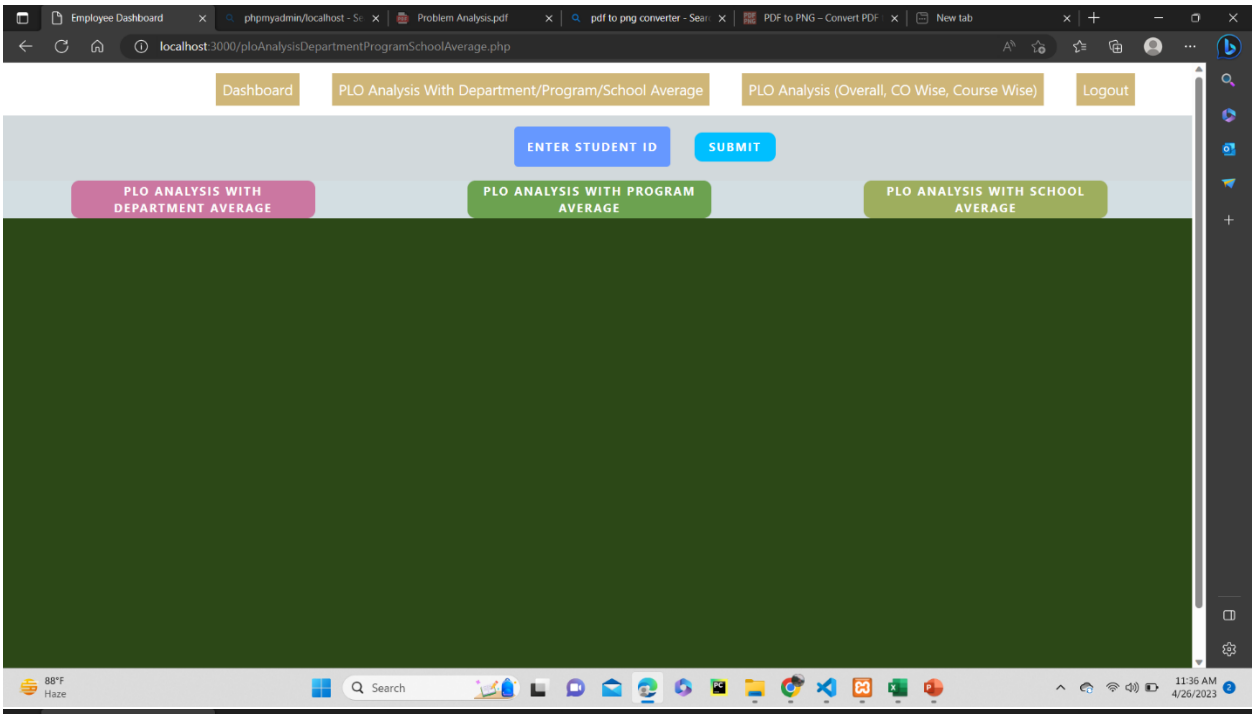
Backlog_data_t

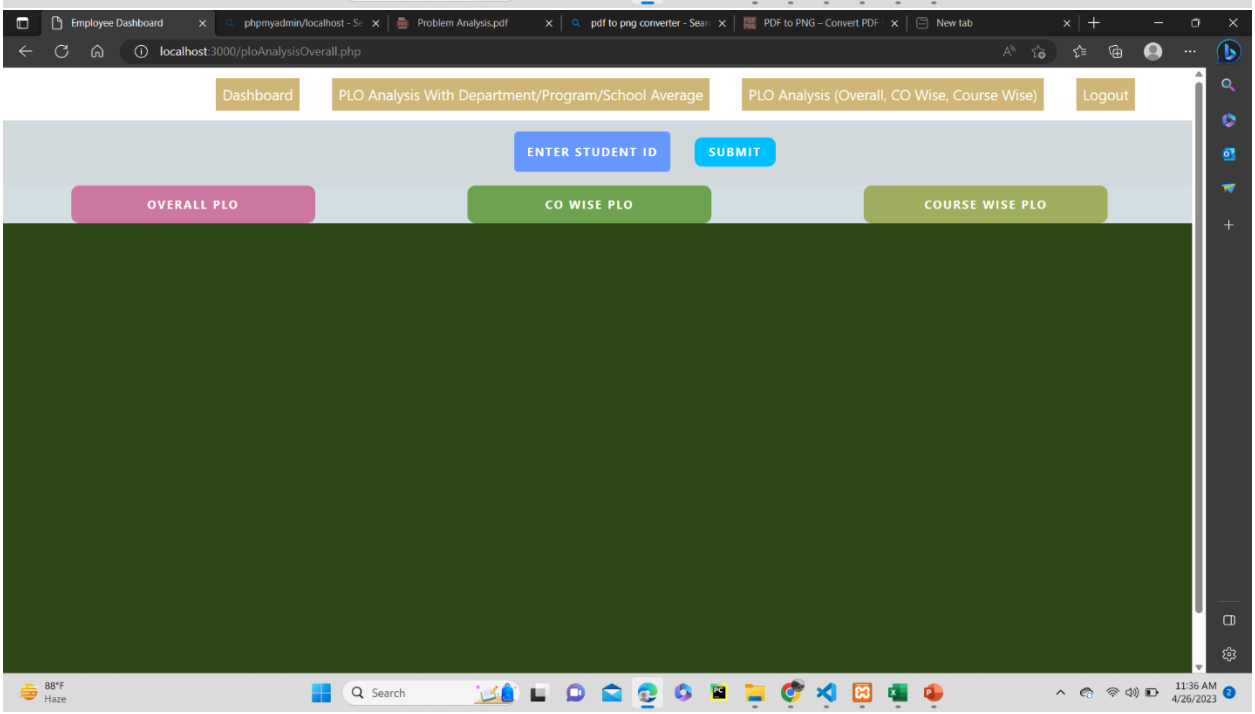
backlogID	INTEGER	11	This is the primary key of backlog_data_t, which is being auto incremented
sectionNum	VARCHAR		This is the section num where students enroll
studentID	INTEGER	11	This is the student ID and it is foreign key, which comes from student table
semester	VARCHAR	6	This is the semester like spring, summer , autumn
courseID	VARCHAR	6	This is the foreign key , comes from course table
facultyID	INTEGER	11	This is the foreign key which comes from the faculty table, it indicates the faculty

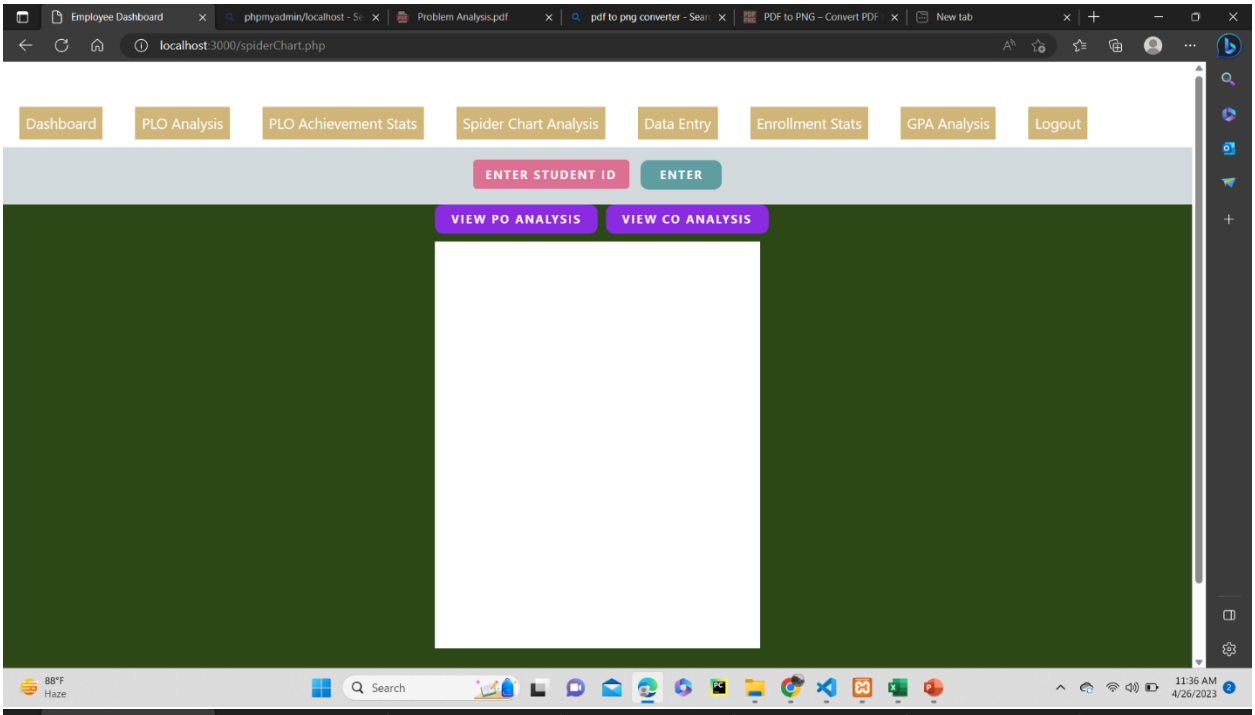
year	YEAR	4	This is the year when the students enrolled in the university
totalMarksObtained	INTEGER	11	This the mark of student , which is being inputed by faculty in this table
time_stamp	TIME_STAMP		This is time stamp; it is being auto stored in the system. When faculty insert the data in the system, it is storing the time and date of that moment in the system

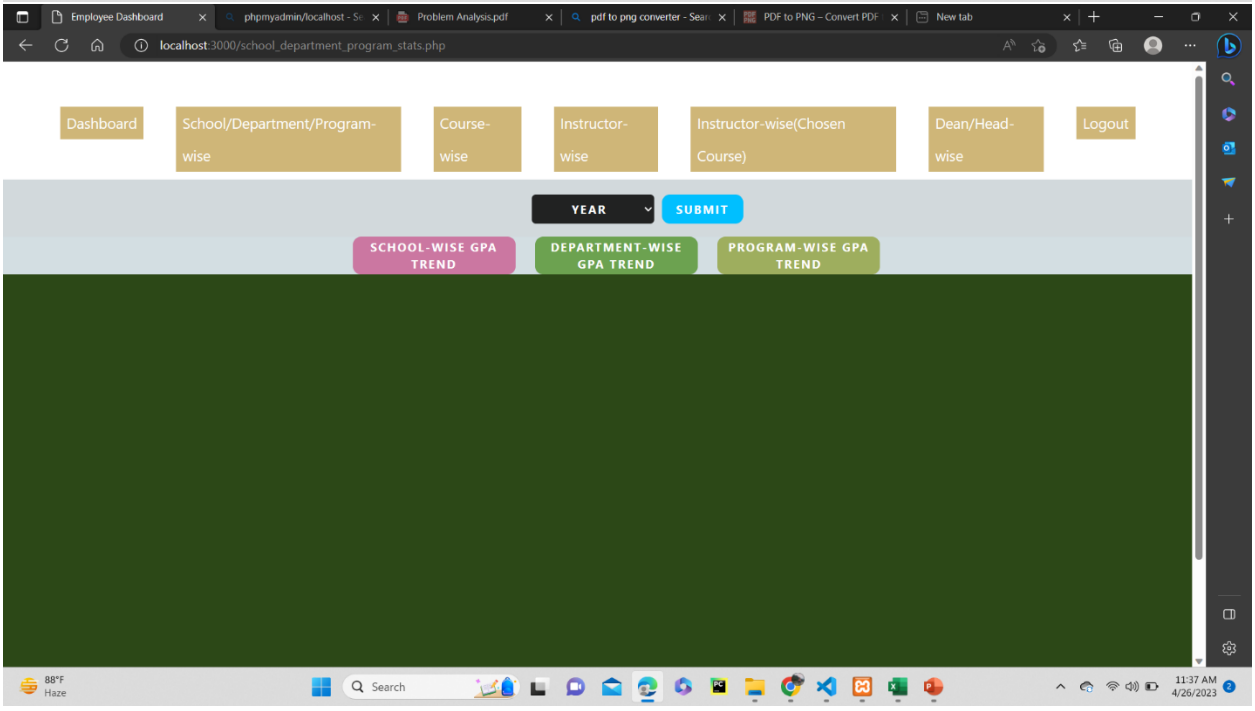
CHAPTER-4 PHYSICAL SYSTEM DESIGN

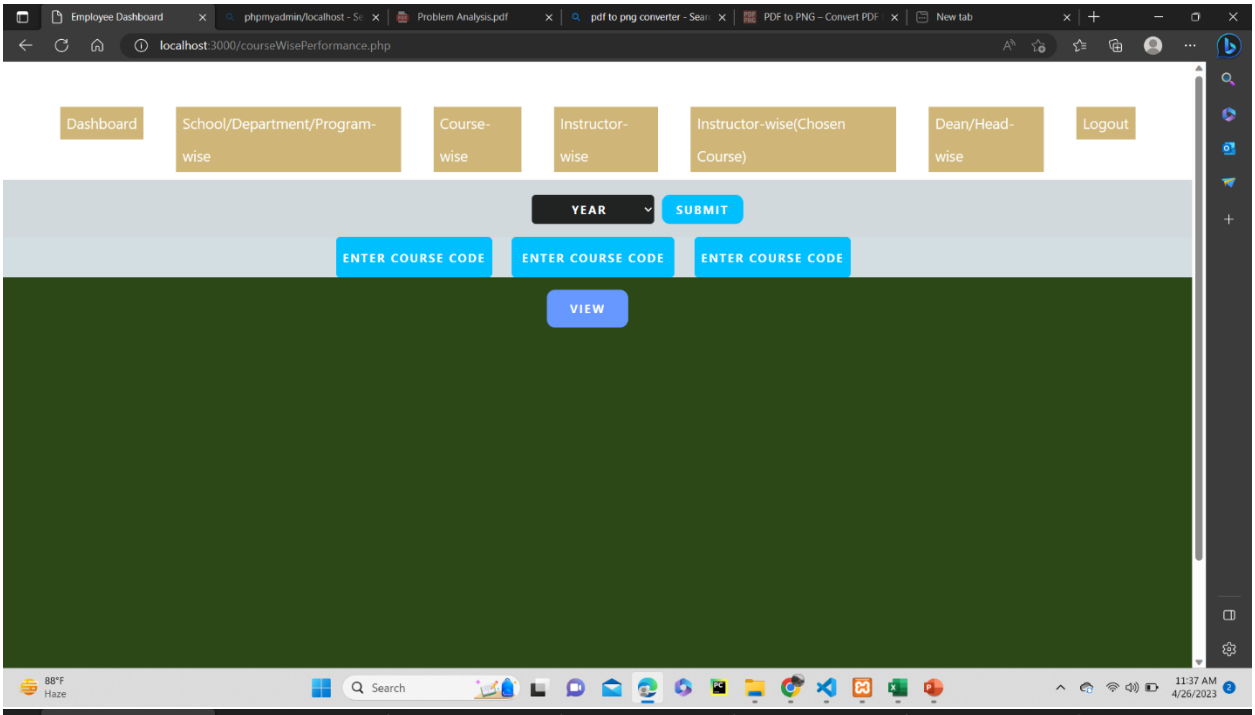
A. INPUT FORMS:

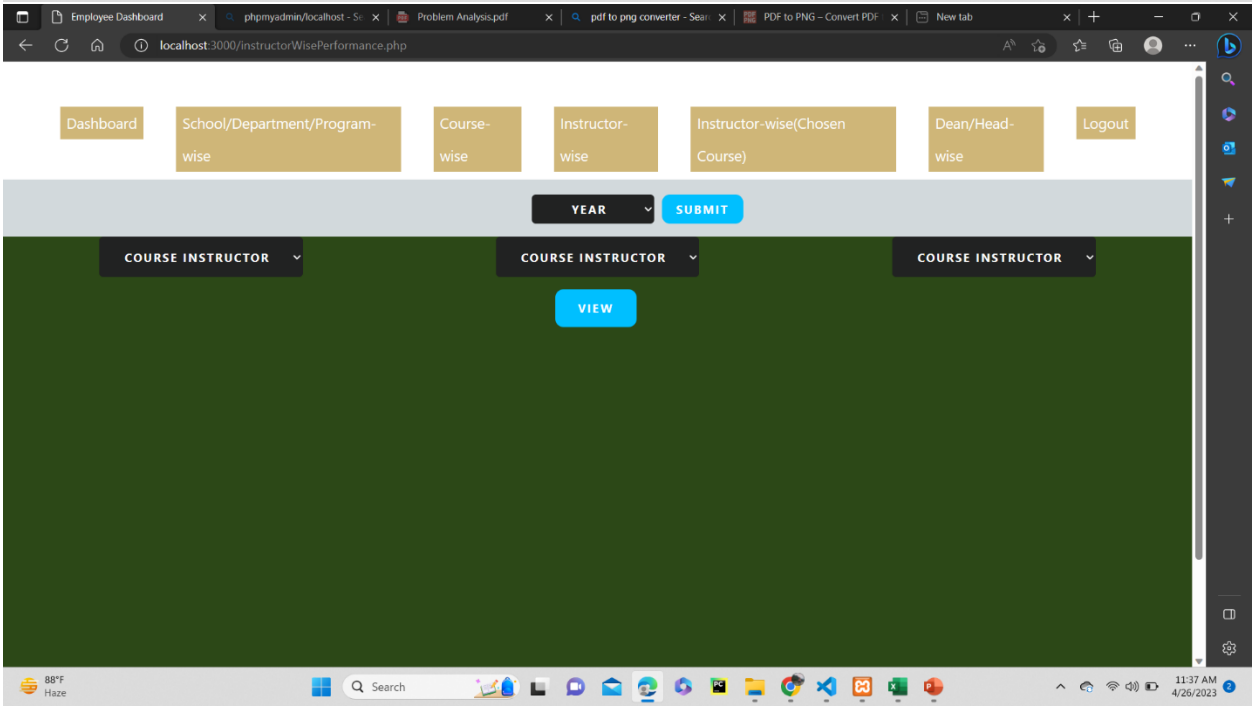


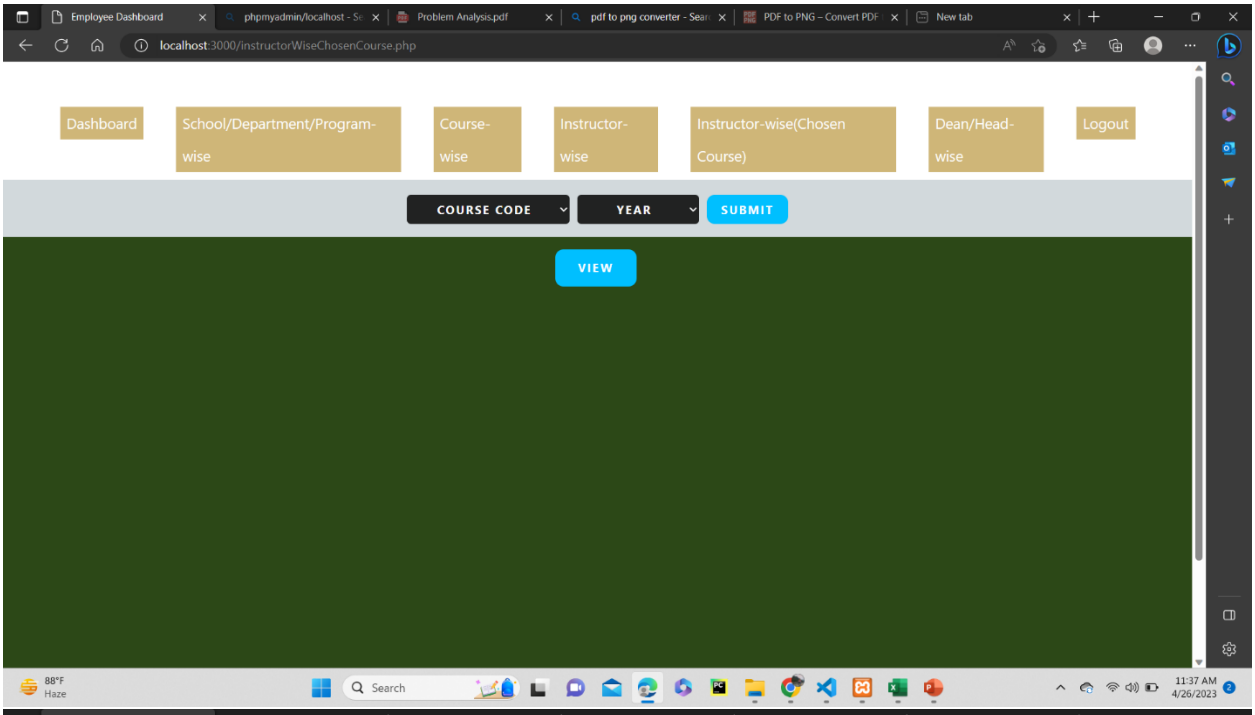


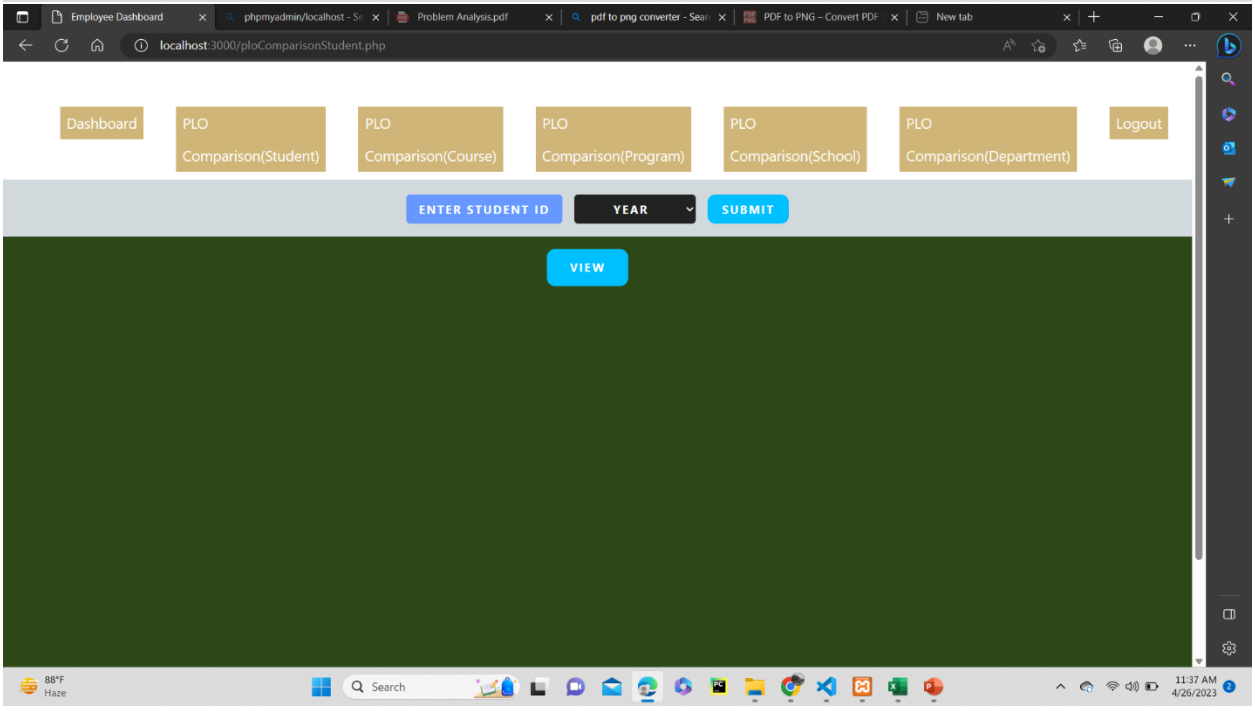

Employee Dashboard
localhost:3000/spiderChart.php
Dashboard
PLO Analysis
PLO Achievement Stats
Spider Chart Analysis
Data Entry
Enrollment Stats
GPA Analysis
Logout
ENTER STUDENT ID
ENTER
VIEW PO ANALYSIS
VIEW CO ANALYSIS

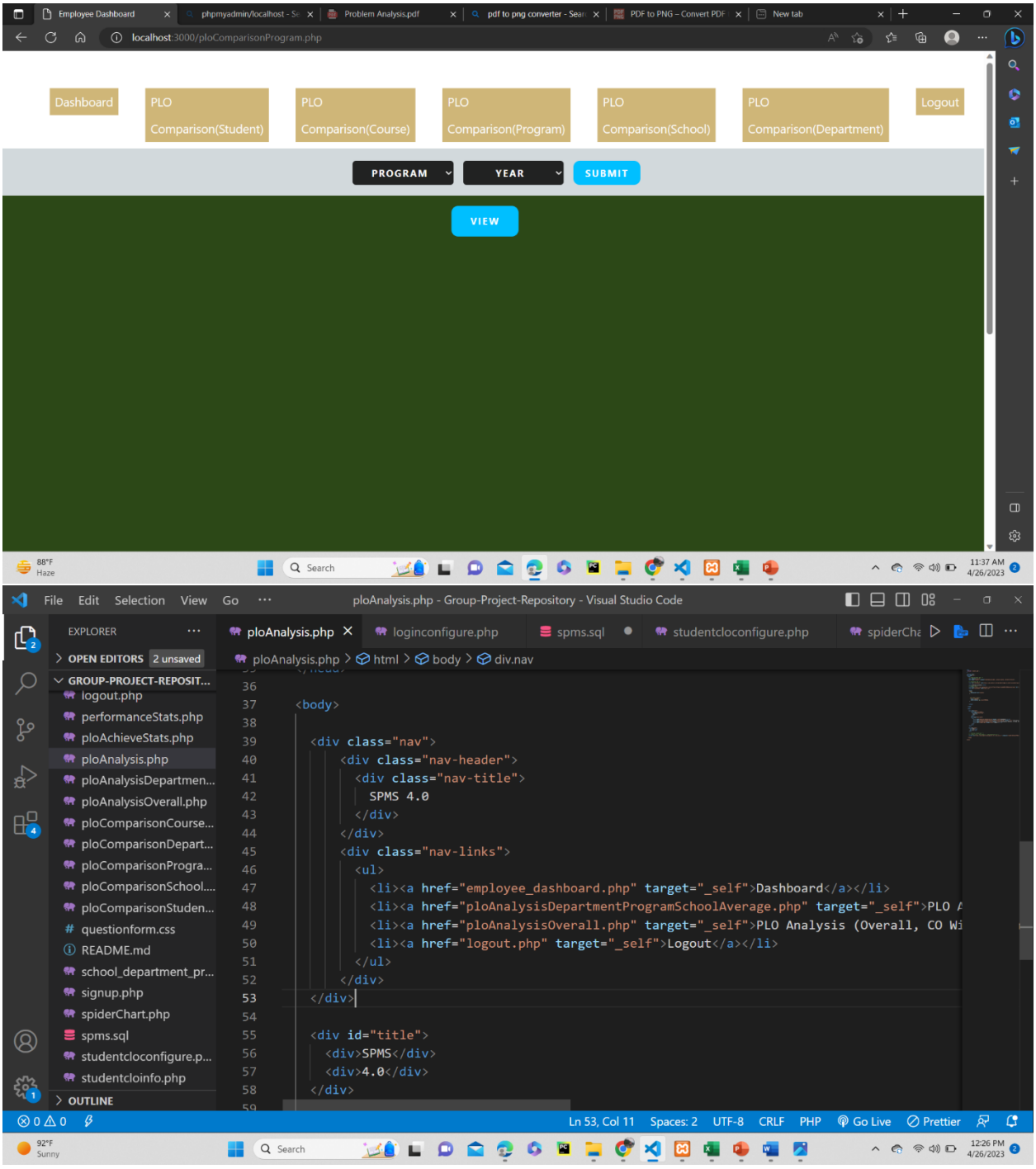

Employee Dashboard
localhost:3000/school_department_program_stats.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
YEAR
SUBMIT
SCHOOL-WISE GPA TREND
DEPARTMENT-WISE GPA TREND
PROGRAM-WISE GPA TREND


Employee Dashboard
localhost:3000/courseWisePerformance.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
YEAR
SUBMIT
ENTER COURSE CODE
ENTER COURSE CODE
ENTER COURSE CODE
VIEW


Employee Dashboard
localhost:3000/instructorWisePerformance.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
YEAR
SUBMIT
COURSE INSTRUCTOR
COURSE INSTRUCTOR
COURSE INSTRUCTOR
VIEW


Employee Dashboard
localhost:3000/instructorWiseChosenCourse.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
COURSE CODE
YEAR
SUBMIT
VIEW


Employee Dashboard
localhost:3000/ploComparisonStudent.php
Dashboard
PLO Comparison(Student)
PLO Comparison(Course)
PLO Comparison(Program)
PLO Comparison(School)
PLO Comparison(Department)
Logout
ENTER STUDENT ID
YEAR
SUBMIT
VIEW

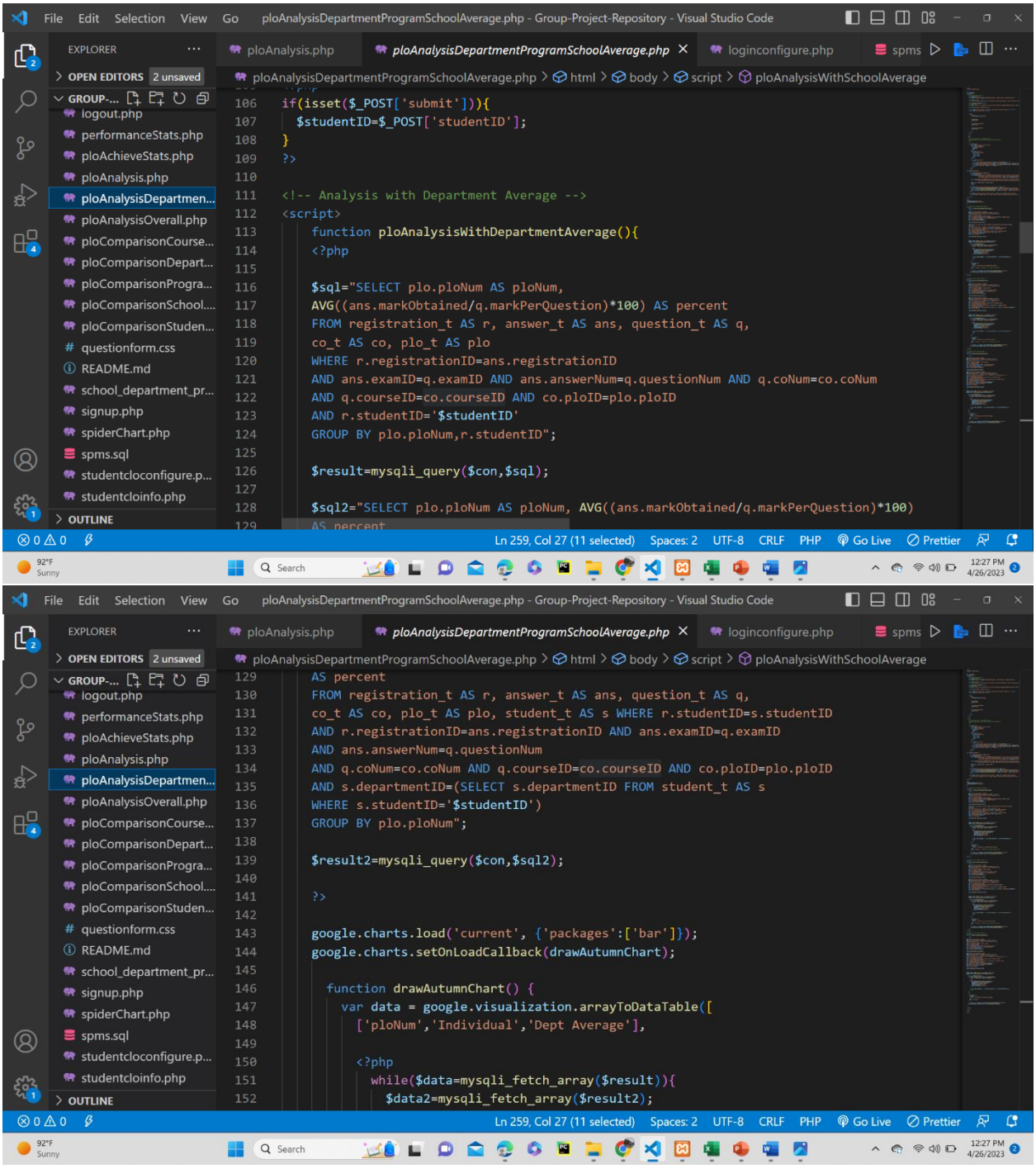

Employee Dashboard
localhost:3000/ploComparisonProgram.php
Dashboard
PLO Comparison(Student)
PLO Comparison(Course)
PLO Comparison(Program)
PLO Comparison(School)
PLO Comparison(Department)
Logout
PROGRAM
YEAR
SUBMIT
VIEW
ploAnalysis.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-PROJECT-REPOSIT...
ploAnalysis.php
loginconfigure.php
spms.sql
studentcloconfigure.php
ploAnalysis.php > html > body > div.nav
<body>

SPMS 4.0

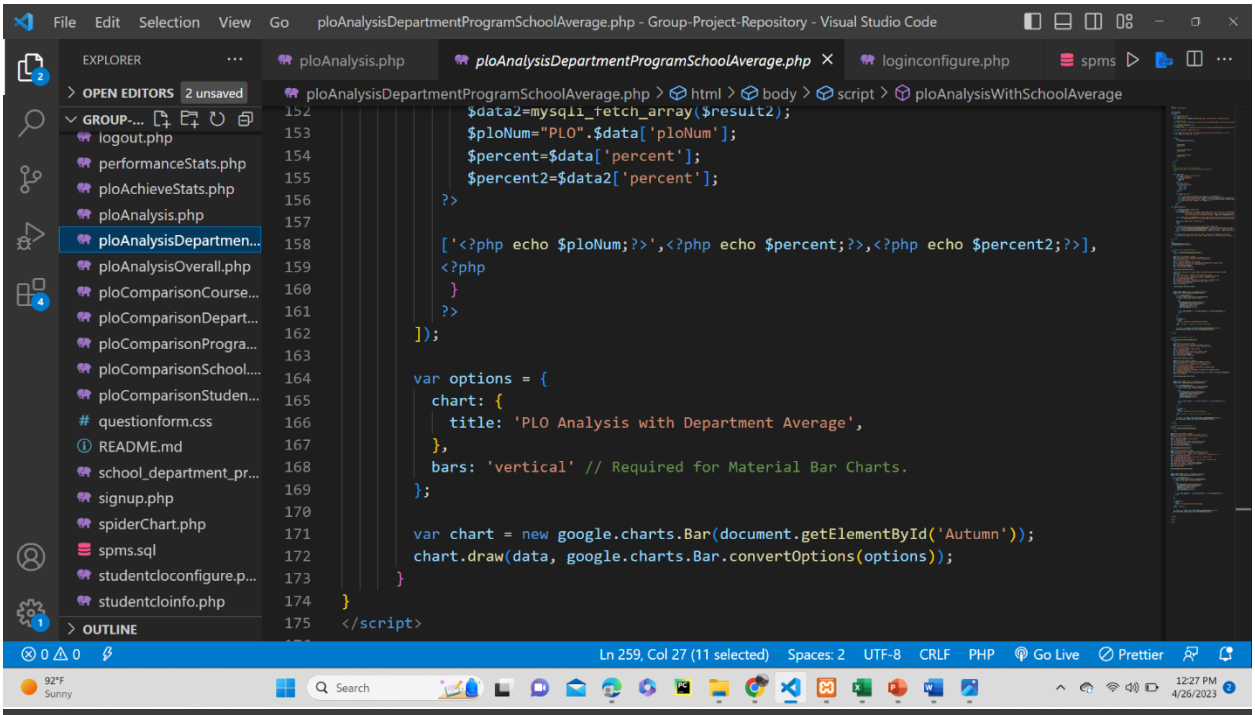
Dashboard
<a href="ploAnalysisDepartmentProgramSchoolAverage.php" target="_self">PLO A
<a href="ploAnalysisOverall.php" target="_self">PLO Analysis (Overall, CO Wi
Logout

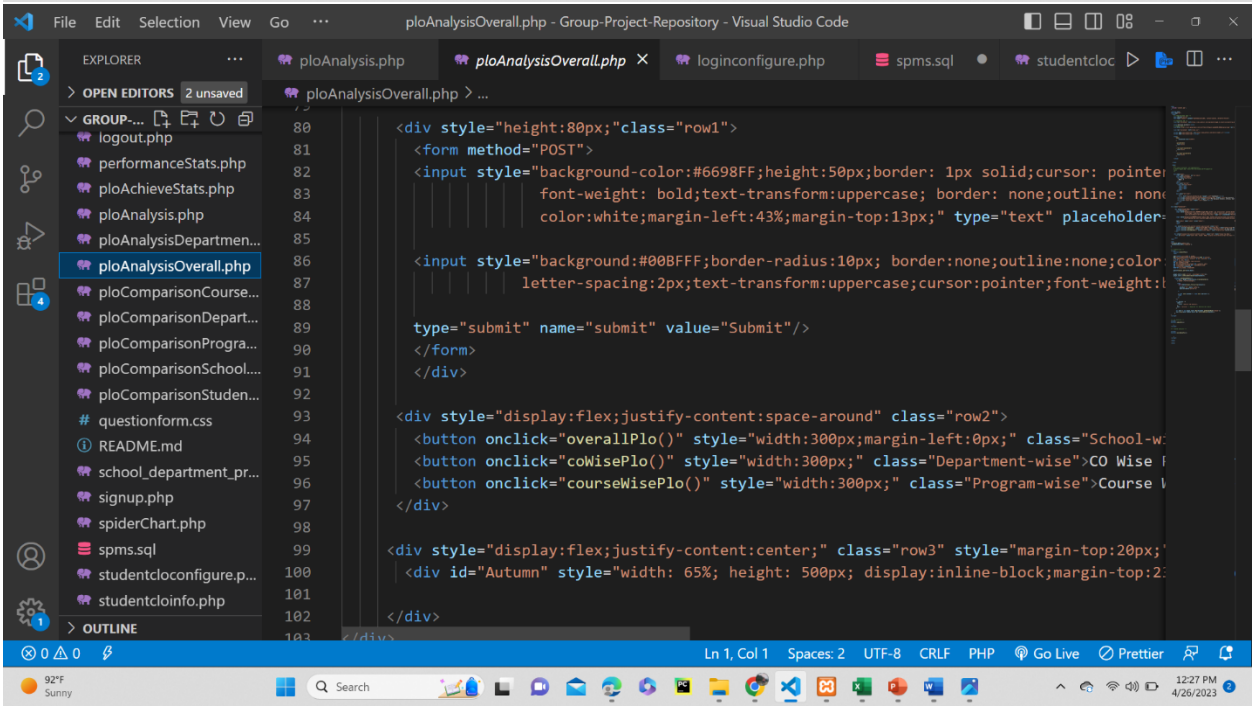
SPMS
4.0

Ln 53, Col 11
Spaces: 2
UTF-8
CRLF
PHP
Go Live
Prettier



```
106  if(isset($_POST['submit'])){
107    $studentID=$_POST['studentID'];
108  }
109  ?>
110
111  <!-- Analysis with Department Average -->
112  <script>
113      function ploAnalysisWithDepartmentAverage(){
114      <?php
115
116      $sql="SELECT plo.ploNum AS ploNum,
117      AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
118      FROM registration_t AS r, answer_t AS ans, question_t AS q,
119      co_t AS co, plo_t AS plo
120      WHERE r.registrationID=ans.registrationID
121      AND ans.examID=q.examID AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
122      AND q.courseID=co.courseID AND co.ploID=plo.ploID
123      AND r.studentID='$studentID'
124      GROUP BY plo.ploNum,r.studentID";
125
126      $result=mysqli_query($con,$sql);
127
128      $sql2="SELECT plo.ploNum AS ploNum, AVG((ans.markObtained/q.markPerQuestion)*100)
129      AS percent
130      FROM registration_t AS r, answer_t AS ans, question_t AS q,
131      co_t AS co, plo_t AS plo, student_t AS s WHERE r.studentID=s.studentID
132      AND r.registrationID=ans.registrationID AND ans.examID=q.examID
133      AND ans.answerNum=q.questionNum
134      AND q.coNum=co.coNum AND q.courseID=co.courseID AND co.ploID=plo.ploID
135      AND s.departmentID=(SELECT s.departmentID FROM student_t AS s
136      WHERE s.studentID='$studentID')
137      GROUP BY plo.ploNum";
138
139      $result2=mysqli_query($con,$sql2);
140
141      ?>
142
143      google.charts.load('current', {'packages':['bar']});
144      google.charts.setOnLoadCallback(drawAutumnChart);
145
146        function drawAutumnChart() {
147          var data = google.visualization.arrayToDataTable([
148            ['ploNum','Individual','Dept Average'],
149
150            <?php
151              while($data=mysqli_fetch_array($result)){
152                $data2=mysqli_fetch_array($result2);
```

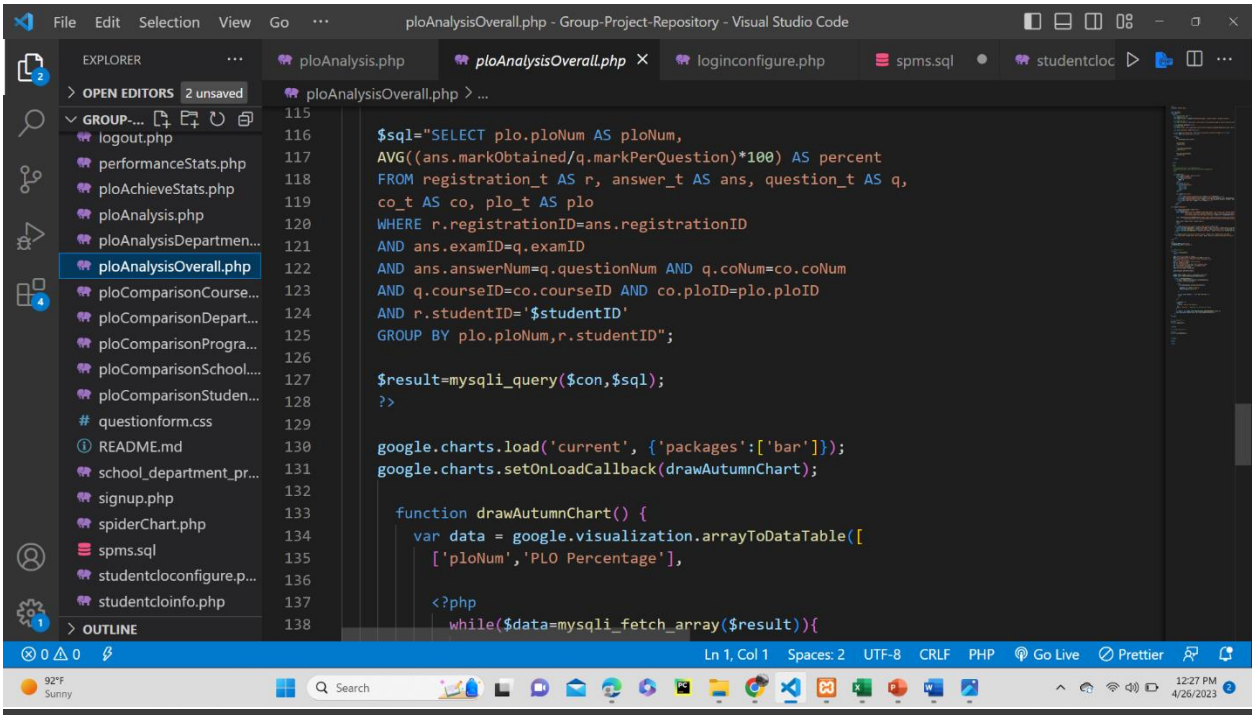

File Edit Selection View Go ploAnalysisDepartmentProgramSchoolAverage.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
logout.php
performanceStats.php
ploAchieveStats.php
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool....
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
OUTLINE
ploAnalysis.php
ploAnalysisDepartmentProgramSchoolAverage.php
loginconfigure.php
spms
ploAnalysisDepartmentProgramSchoolAverage.php > html > body > script > ploAnalysisWithSchoolAverage
$data2=mysqli_fetch_array($result2);
$ploNum="PLO".$data['ploNum'];
$percent=$data['percent'];
$percent2=$data2['percent'];
?>
['<?php echo $ploNum;?>',<?php echo $percent;?>,<?php echo $percent2;?>],
<?php
}
?>
]);
var options = {
chart: {
title: 'PLO Analysis with Department Average',
},
bars: 'vertical' // Required for Material Bar Charts.
};
var chart = new google.charts.Bar(document.getElementById('Autumn'));
chart.draw(data, google.charts.Bar.convertOptions(options));
}
}
</script>
Ln 259, Col 27 (11 selected) Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
92°F Sunny
Search
12:27 PM 4/26/2023

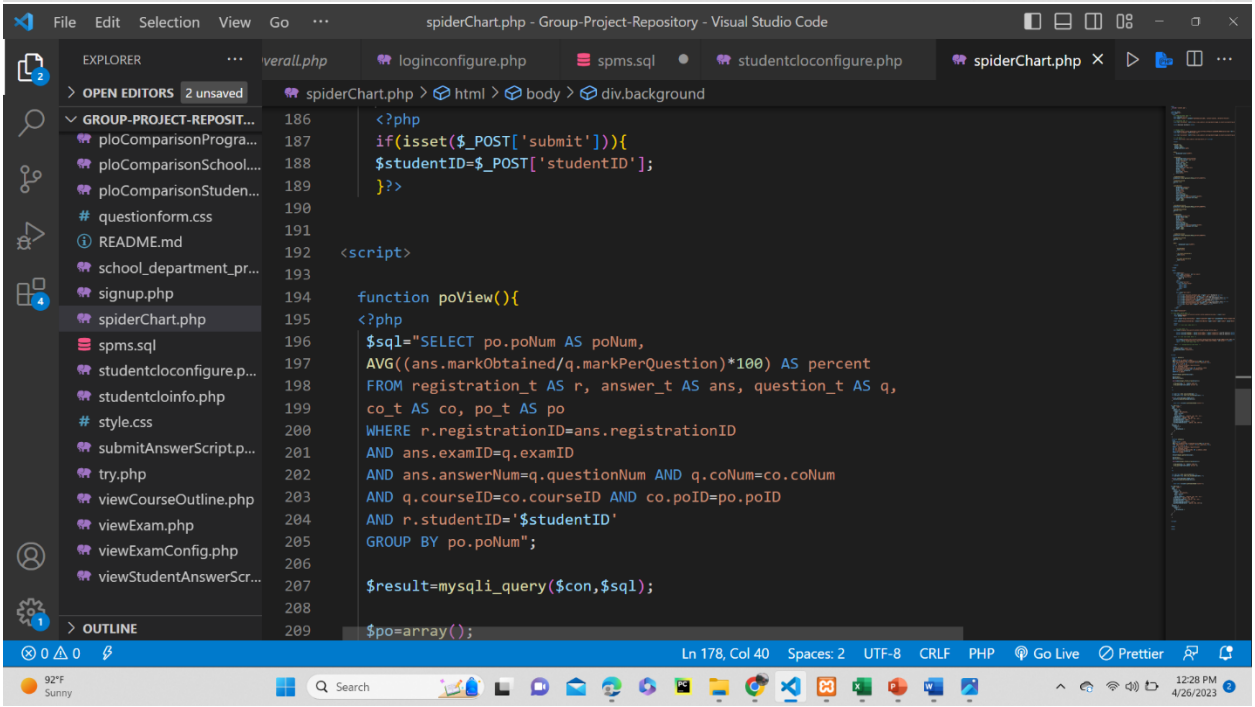

File Edit Selection View Go ploAnalysisOverall.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
logout.php
performanceStats.php
ploAchieveStats.php
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool....
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
OUTLINE
ploAnalysis.php
ploAnalysisOverall.php
loginconfigure.php
spms.sql
studentcloc
ploAnalysisOverall.php > ...
<div style="height:80px;"class="row1">
<form method="POST">
<input style="background-color:#6698FF;height:50px;border: 1px solid;cursor: pointer
font-weight: bold;text-transform:uppercase; border: none;outline: none
color:white;margin-left:43%;margin-top:13px;" type="text" placeholder=
<input style="background:#00BFFF;border-radius:10px; border:none;outline:none;color
letter-spacing:2px;text-transform:uppercase;cursor:pointer;font-weight:b
type="submit" name="submit" value="Submit"/>
</form>

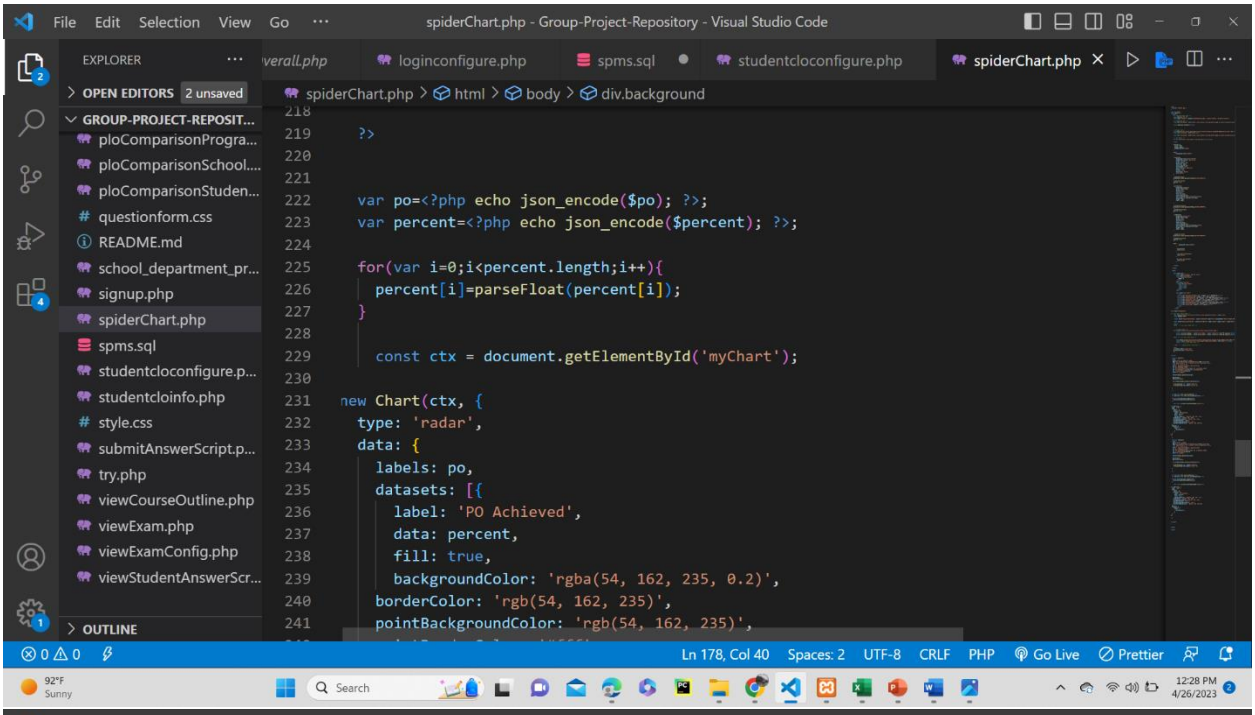
<button onclick="overallPlo()" style="width:300px;margin-left:0px;" class="School-w
<button onclick="coWisePlo()" style="width:300px;" class="Department-wise">CO Wise P
<button onclick="courseWisePlo()" style="width:300px;" class="Program-wise">Course W

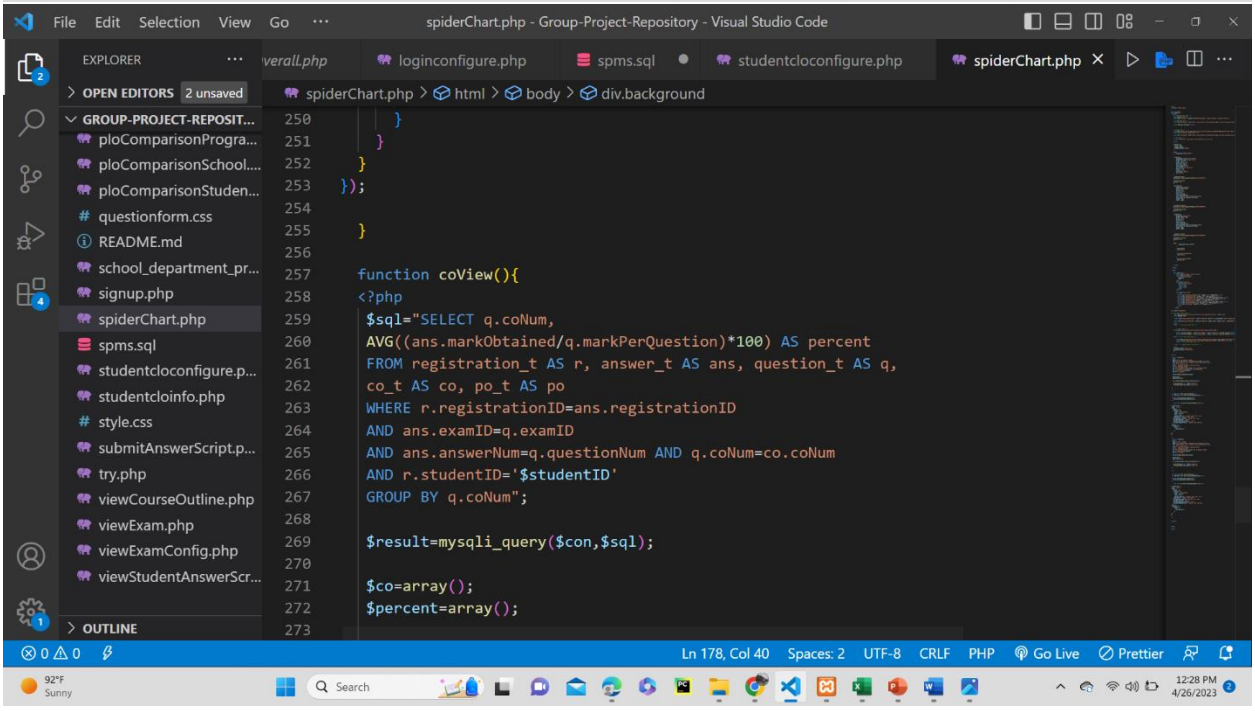
<div style="display:flex;justify-content:center;" class="row3" style="margin-top:20px;
<div id="Autumn" style="width: 65%; height: 500px; display:inline-block;margin-top:2

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
92°F Sunny
Search
12:27 PM 4/26/2023

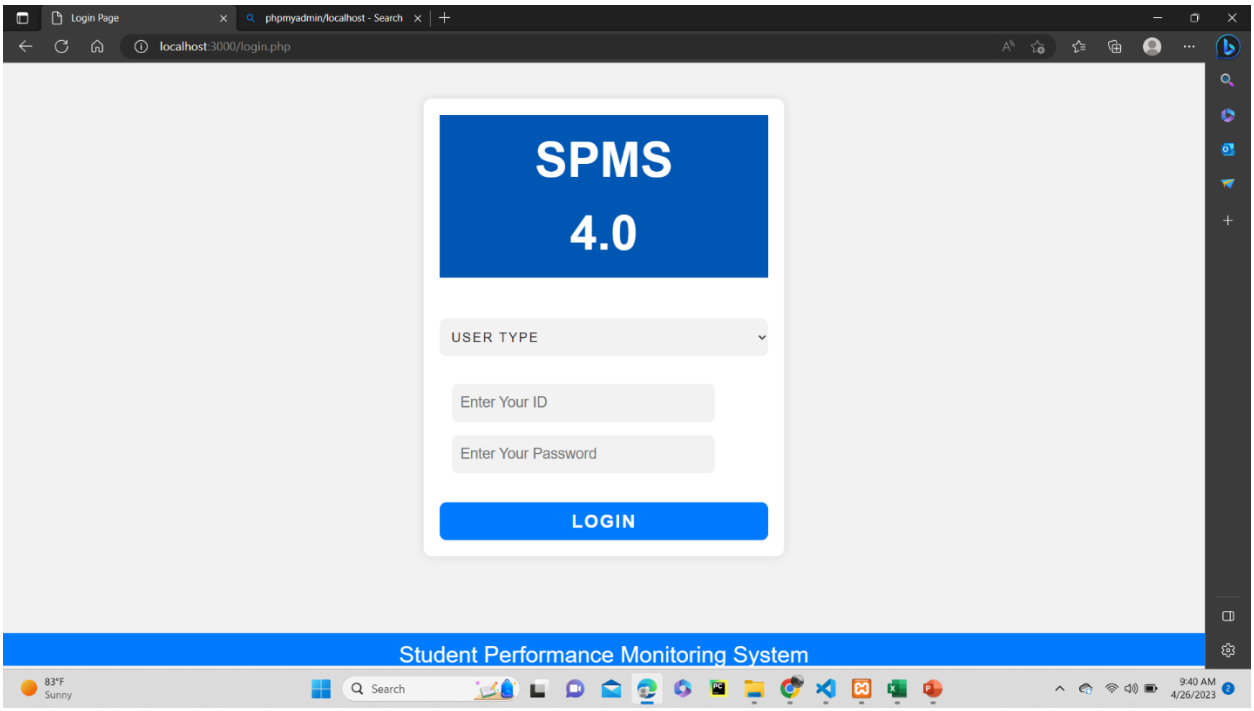

File Edit Selection View Go
ploAnalysisOverall.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
logout.php
performanceStats.php
ploAchieveStats.php
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool....
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
OUTLINE
ploAnalysis.php
ploAnalysisOverall.php
loginconfigure.php
spms.sql
studentcloc
ploAnalysisOverall.php > ...
$sql="SELECT plo.ploNum AS ploNum,
AVG((ans.markObtained/q.markPerQuestion)*100) AS percent
FROM registration_t AS r, answer_t AS ans, question_t AS q,
co_t AS co, plo_t AS plo
WHERE r.registrationID=ans.registrationID
AND ans.examID=q.examID
AND ans.answerNum=q.questionNum AND q.coNum=co.coNum
AND q.courseID=co.courseID AND co.ploID=plo.ploID
AND r.studentID='$studentID'
GROUP BY plo.ploNum,r.studentID";
$result=mysqli_query($con,$sql);
?>
google.charts.load('current', {'packages':['bar']});
google.charts.setOnLoadCallback(drawAutumnChart);
function drawAutumnChart() {
var data = google.visualization.arrayToDataTable([
['ploNum','PLO Percentage'],
<?php
while($data=mysqli_fetch_array($result)){
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
92°F Sunny
Search
12:27 PM 4/26/2023

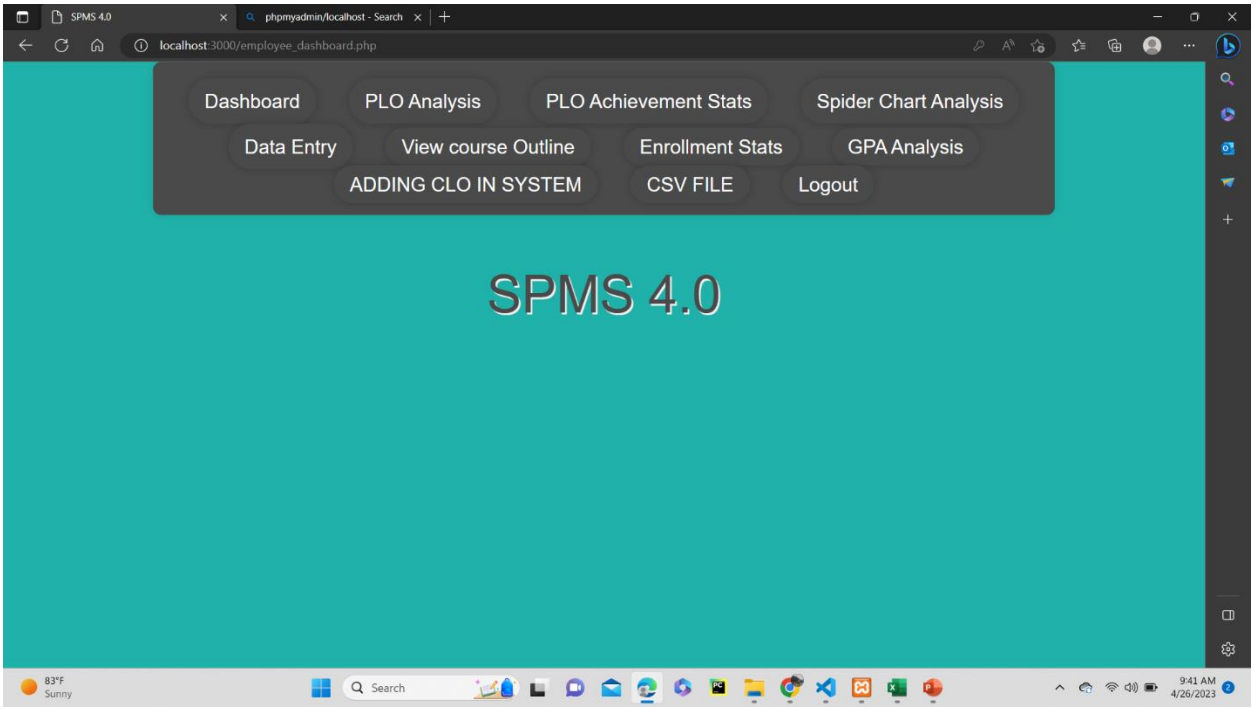

File Edit Selection View Go
spiderChart.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-PROJECT-REPOSIT...
ploComparisonProgra...
ploComparisonSchool....
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
viewCourseOutline.php
viewExam.php
viewExamConfig.php
viewStudentAnswerScr...
OUTLINE
loginconfigure.php
spms.sql
studentcloconfigure.php
spiderChart.php
spiderChart.php > html > body > div.background
<?php
if(isset($_POST['submit'])){
$studentID=$_POST['studentID'];
}?>
<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();
Ln 178, Col 40 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
92°F Sunny
Search
12:28 PM 4/26/2023

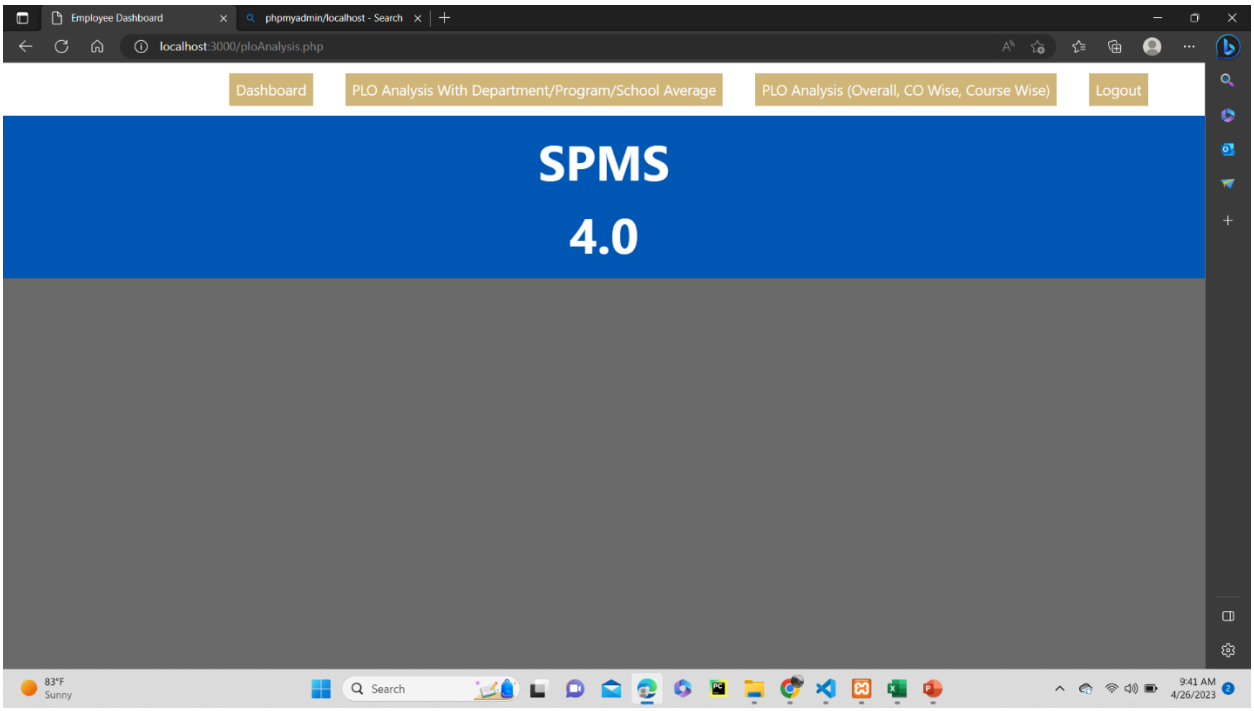

File Edit Selection View Go
spiderChart.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-PROJECT-REPOSIT...
loginconfigure.php
spms.sql
studentcloconfigure.php
spiderChart.php
spiderChart.php > html > body > div.background
OUTLINE
Ln 178, Col 40 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier

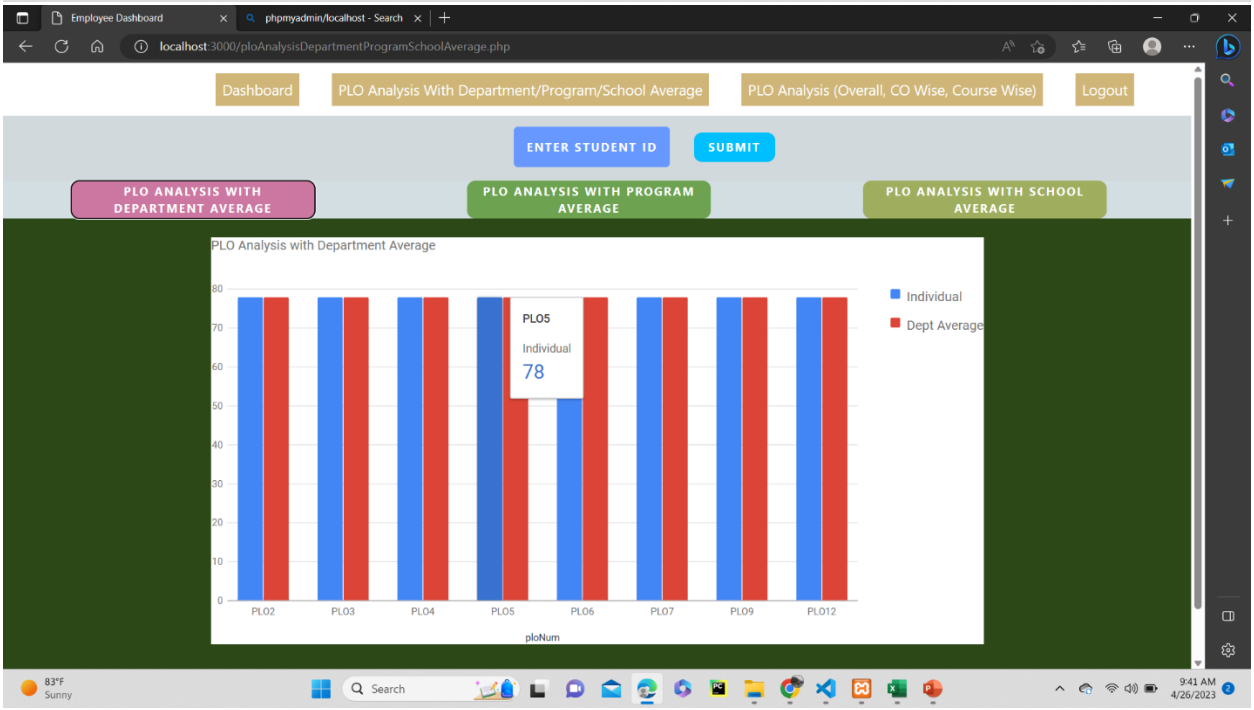

File Edit Selection View Go
spiderChart.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-PROJECT-REPOSIT...
loginconfigure.php
spms.sql
studentcloconfigure.php
spiderChart.php
spiderChart.php > html > body > div.background
OUTLINE
Ln 178, Col 40 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier

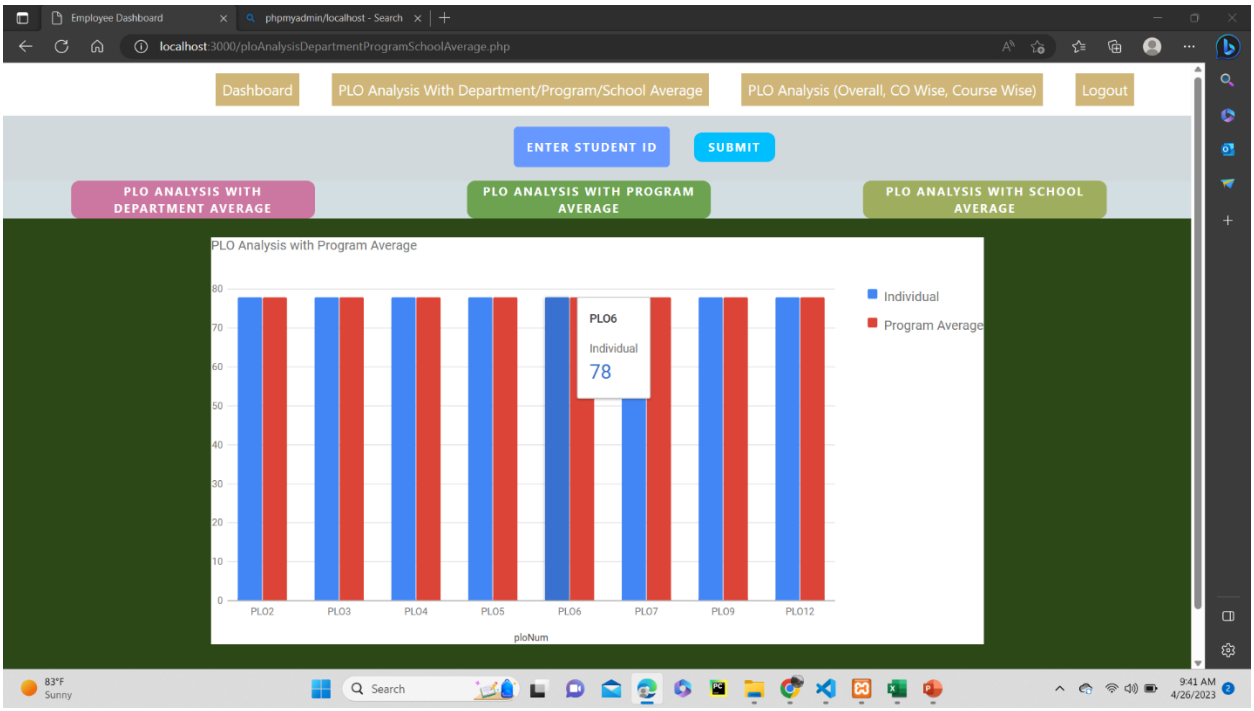
B. OUTPUT FORMS:

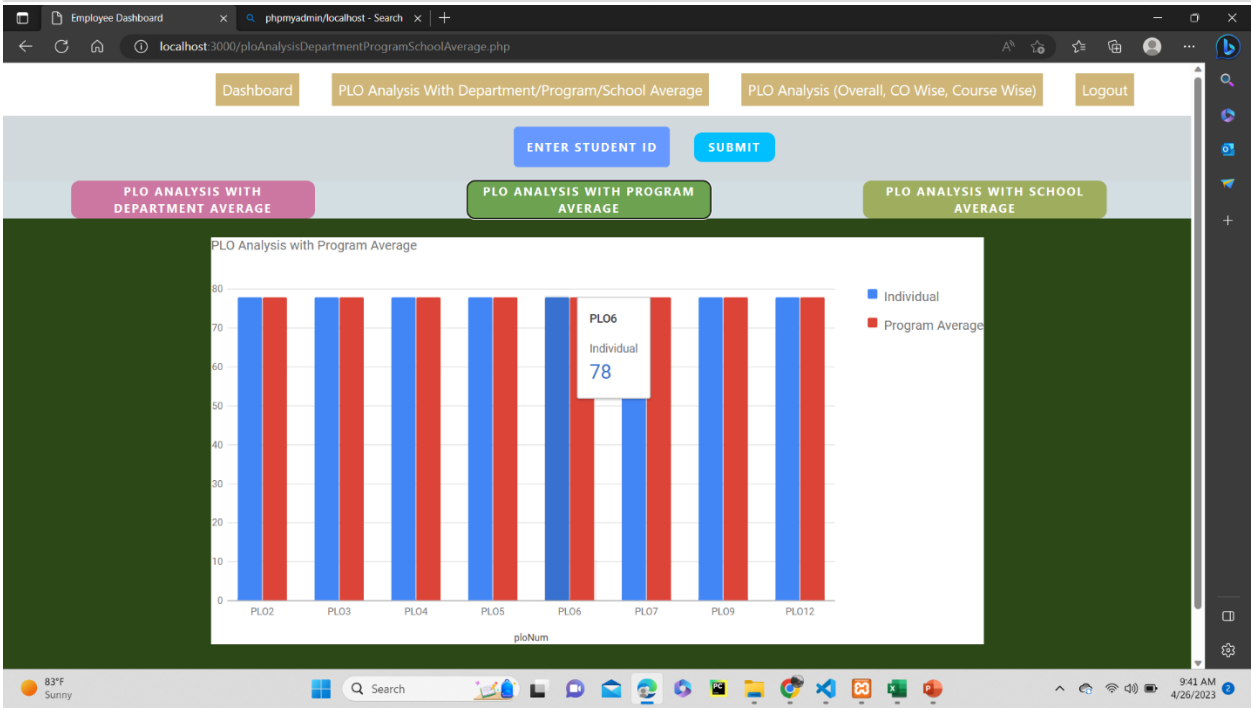


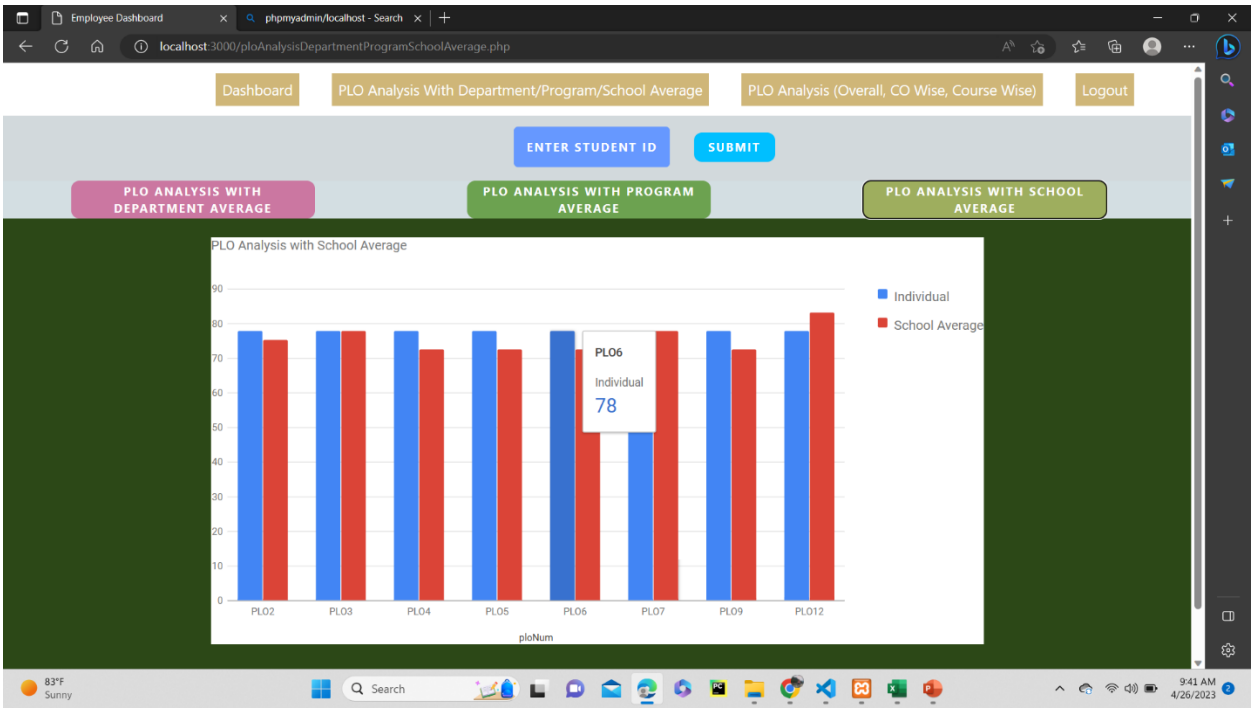


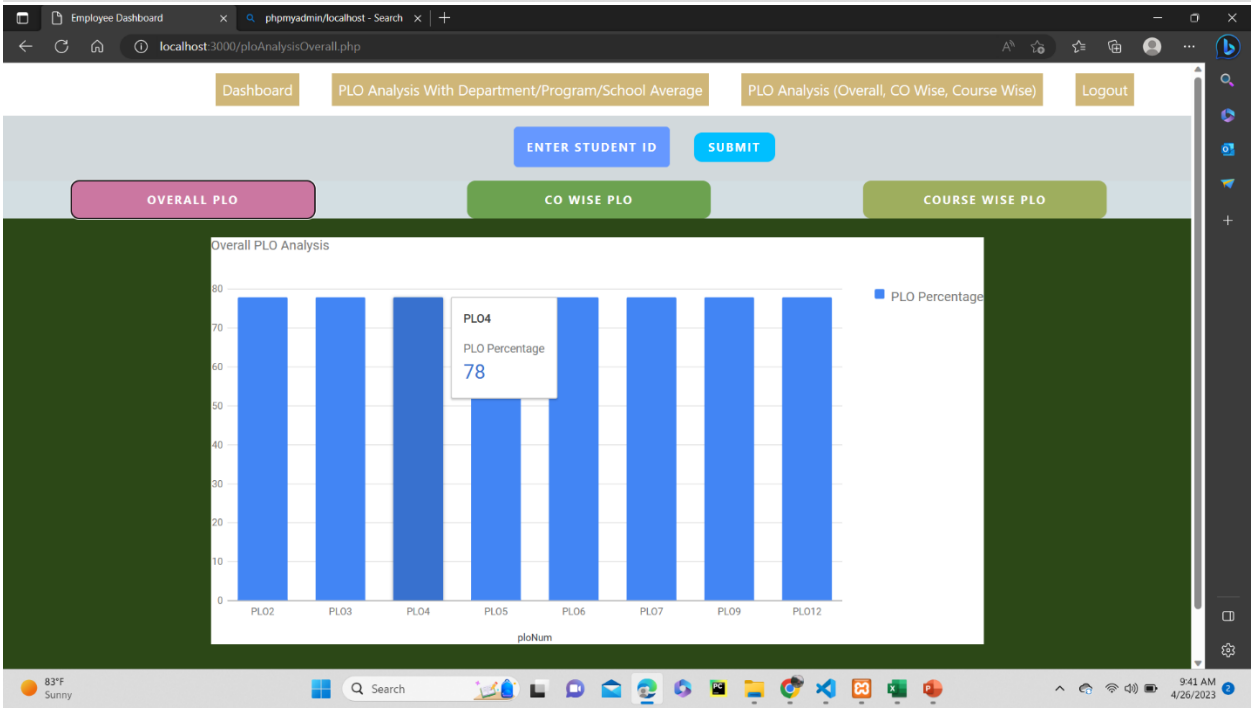

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/ploAnalysis.php
Dashboard
PLO Analysis With Department/Program/School Average
PLO Analysis (Overall, CO Wise, Course Wise)
Logout
SPMS
4.0
Search
9:41 AM
4/26/2023

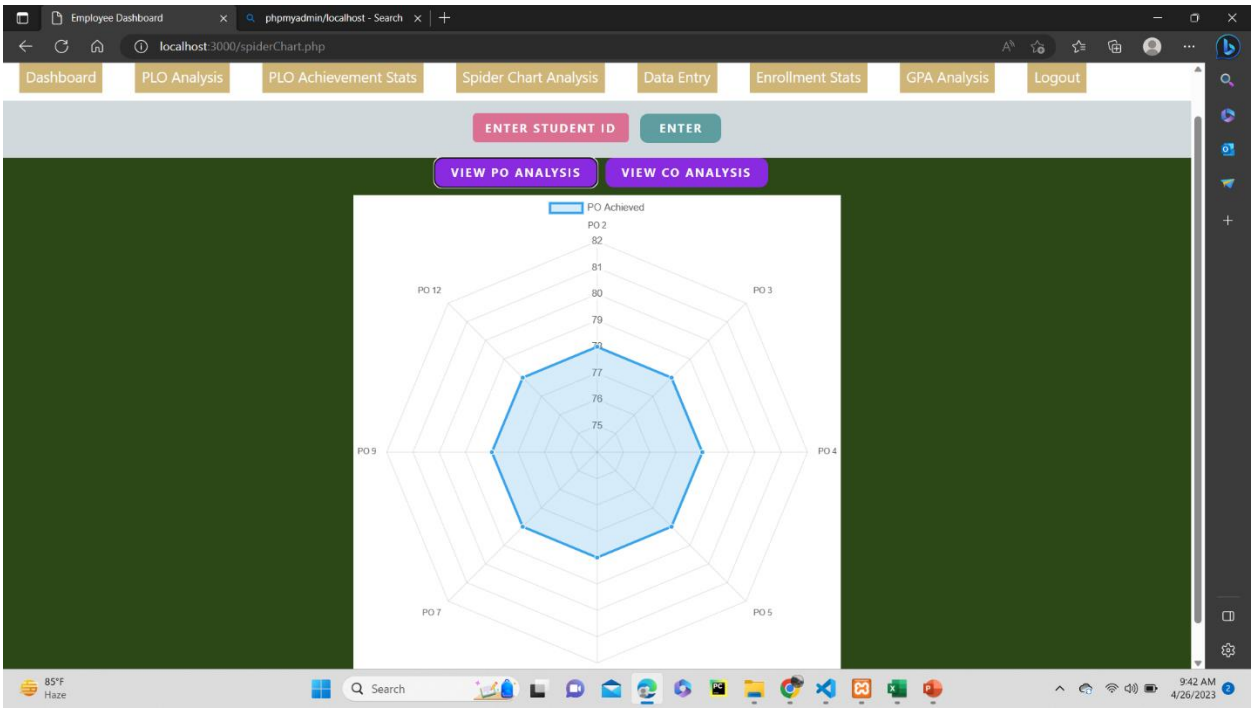

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/ploAnalysisDepartmentProgramSchoolAverage.php
Dashboard
PLO Analysis With Department/Program/School Average
PLO Analysis (Overall, CO Wise, Course Wise)
Logout
ENTER STUDENT ID
SUBMIT
PLO ANALYSIS WITH DEPARTMENT AVERAGE
PLO ANALYSIS WITH PROGRAM AVERAGE
PLO ANALYSIS WITH SCHOOL AVERAGE
PLO Analysis with Department Average
PLO5
Individual
78
Individual
Dept Average
PLO2
PLO3
PLO4
PLO5
PLO6
PLO7
PLO9
PLO12
ploNum
Search
9:41 AM
4/26/2023

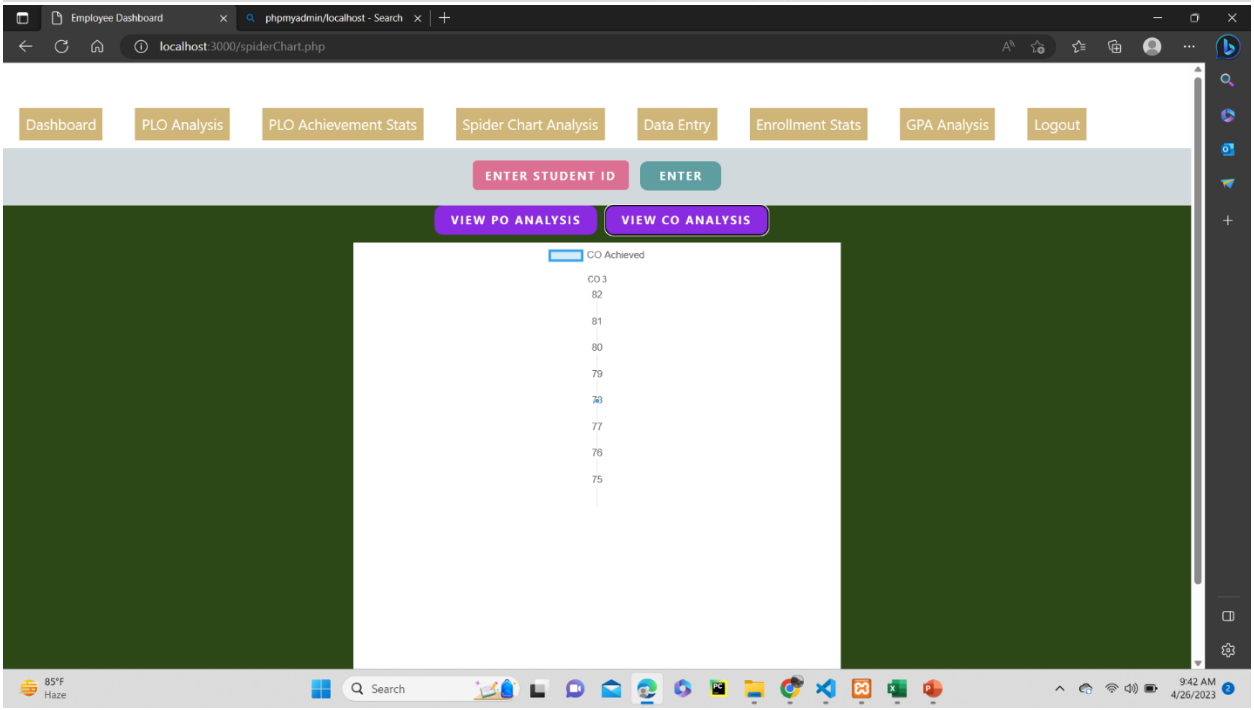

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/ploAnalysisDepartmentProgramSchoolAverage.php
Dashboard
PLO Analysis With Department/Program/School Average
PLO Analysis (Overall, CO Wise, Course Wise)
Logout
ENTER STUDENT ID
SUBMIT
PLO ANALYSIS WITH DEPARTMENT AVERAGE
PLO ANALYSIS WITH PROGRAM AVERAGE
PLO ANALYSIS WITH SCHOOL AVERAGE
PLO Analysis with Program Average
PL06
Individual
78
Individual
Program Average
PL02
PL03
PL04
PL05
PL06
PL07
PL09
PL012
ploNum
83°F
Sunny
Search
9:41 AM
4/26/2023

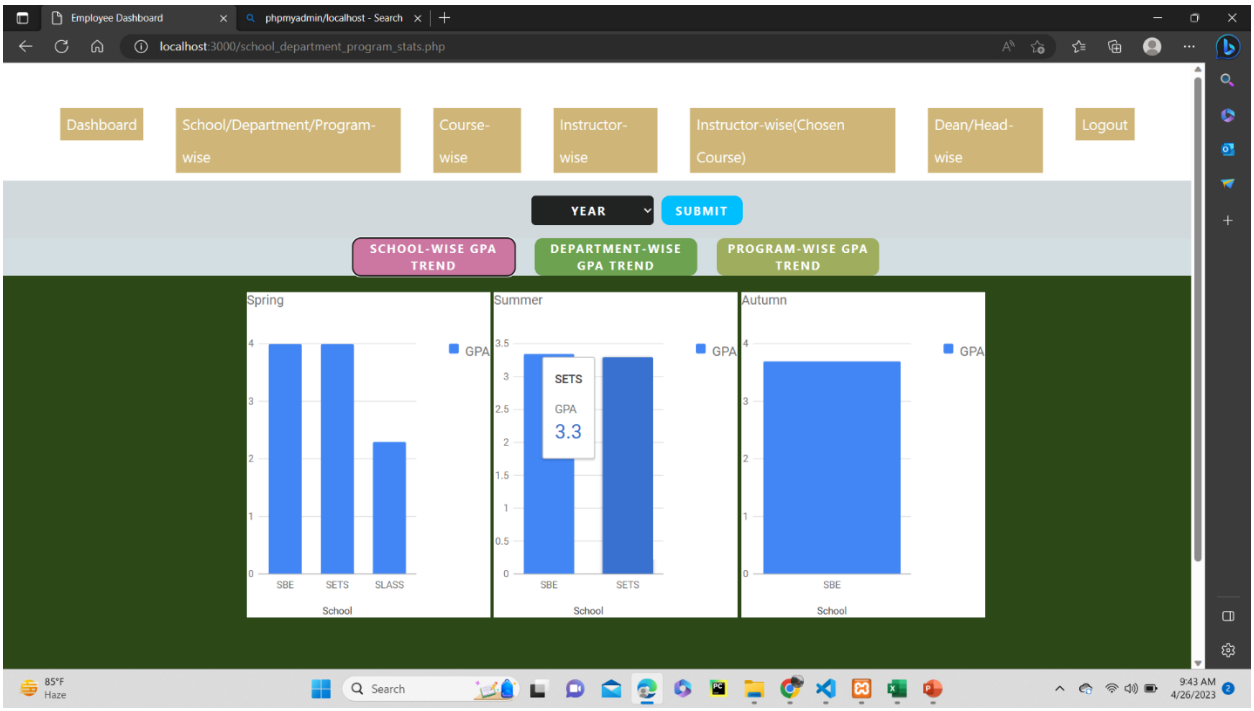

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/ploAnalysisDepartmentProgramSchoolAverage.php
Dashboard
PLO Analysis With Department/Program/School Average
PLO Analysis (Overall, CO Wise, Course Wise)
Logout
ENTER STUDENT ID
SUBMIT
PLO ANALYSIS WITH DEPARTMENT AVERAGE
PLO ANALYSIS WITH PROGRAM AVERAGE
PLO ANALYSIS WITH SCHOOL AVERAGE
PLO Analysis with Program Average
PL06
Individual
78
Individual
Program Average
PL02
PL03
PL04
PL05
PL06
PL07
PL09
PL012
ploNum
83°F
Sunny
Search
9:41 AM
4/26/2023

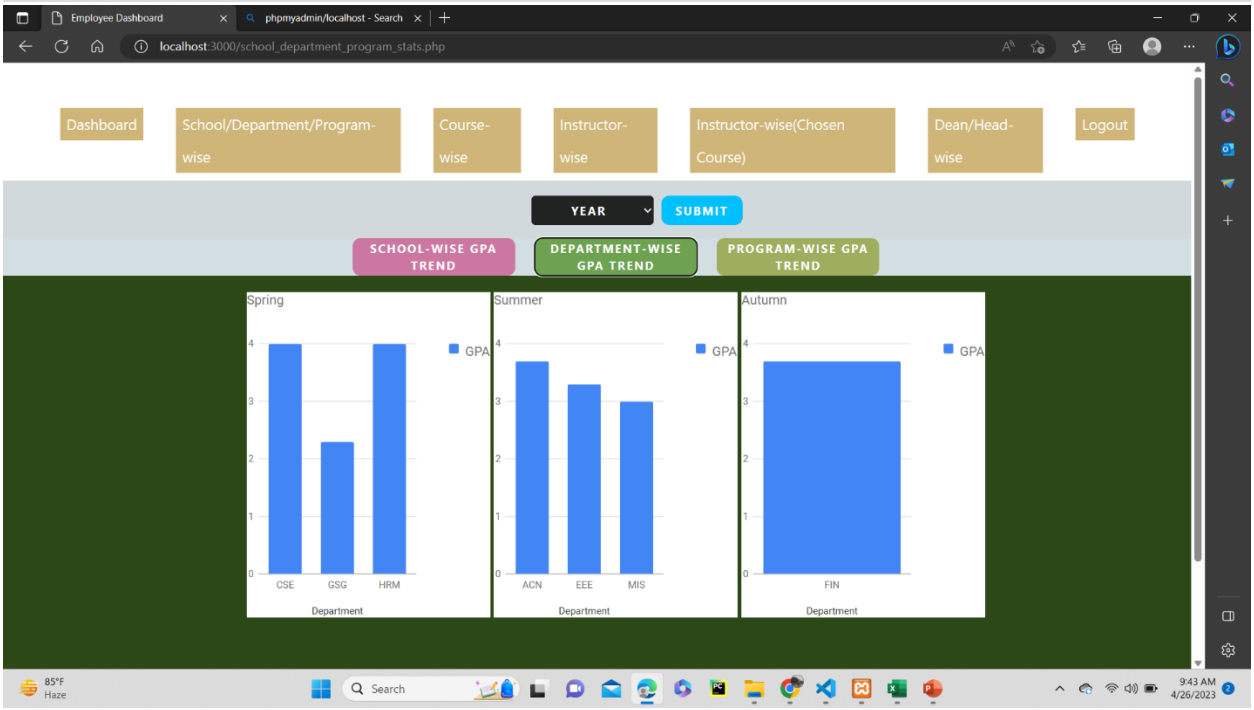

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/ploAnalysisDepartmentProgramSchoolAverage.php
Dashboard
PLO Analysis With Department/Program/School Average
PLO Analysis (Overall, CO Wise, Course Wise)
Logout
ENTER STUDENT ID
SUBMIT
PLO ANALYSIS WITH DEPARTMENT AVERAGE
PLO ANALYSIS WITH PROGRAM AVERAGE
PLO ANALYSIS WITH SCHOOL AVERAGE
PLO Analysis with School Average
Individual
School Average
PL06
Individual
78
PL02
PL03
PL04
PL05
PL06
PL07
PL09
PL012
ploNum
83°F
Sunny
Search
9:41 AM
4/26/2023

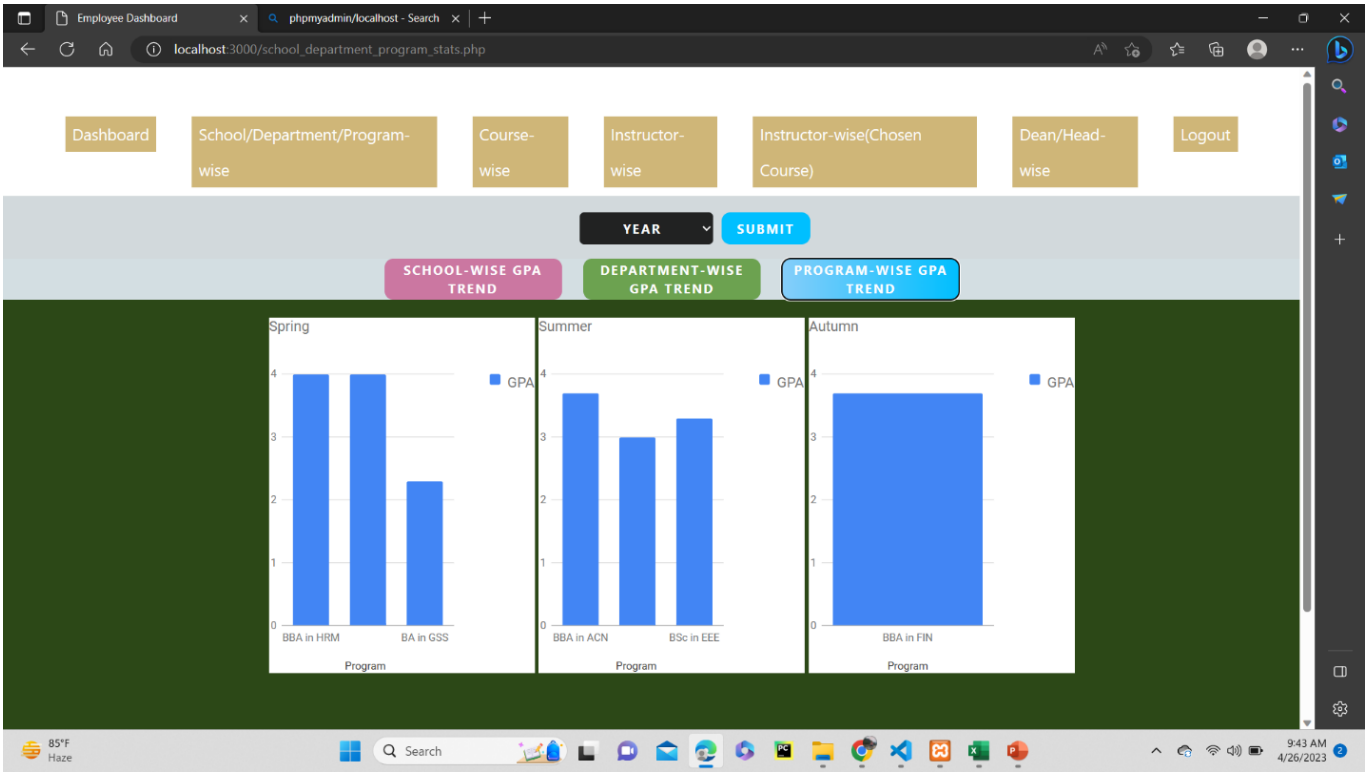

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/ploAnalysisOverall.php
Dashboard
PLO Analysis With Department/Program/School Average
PLO Analysis (Overall, CO Wise, Course Wise)
Logout
ENTER STUDENT ID
SUBMIT
OVERALL PLO
CO WISE PLO
COURSE WISE PLO
Overall PLO Analysis
PLO Percentage
PL04
PLO Percentage
78
PL02
PL03
PL04
PL05
PL06
PL07
PL09
PL012
ploNum
83°F
Sunny
Search
9:41 AM
4/26/2023

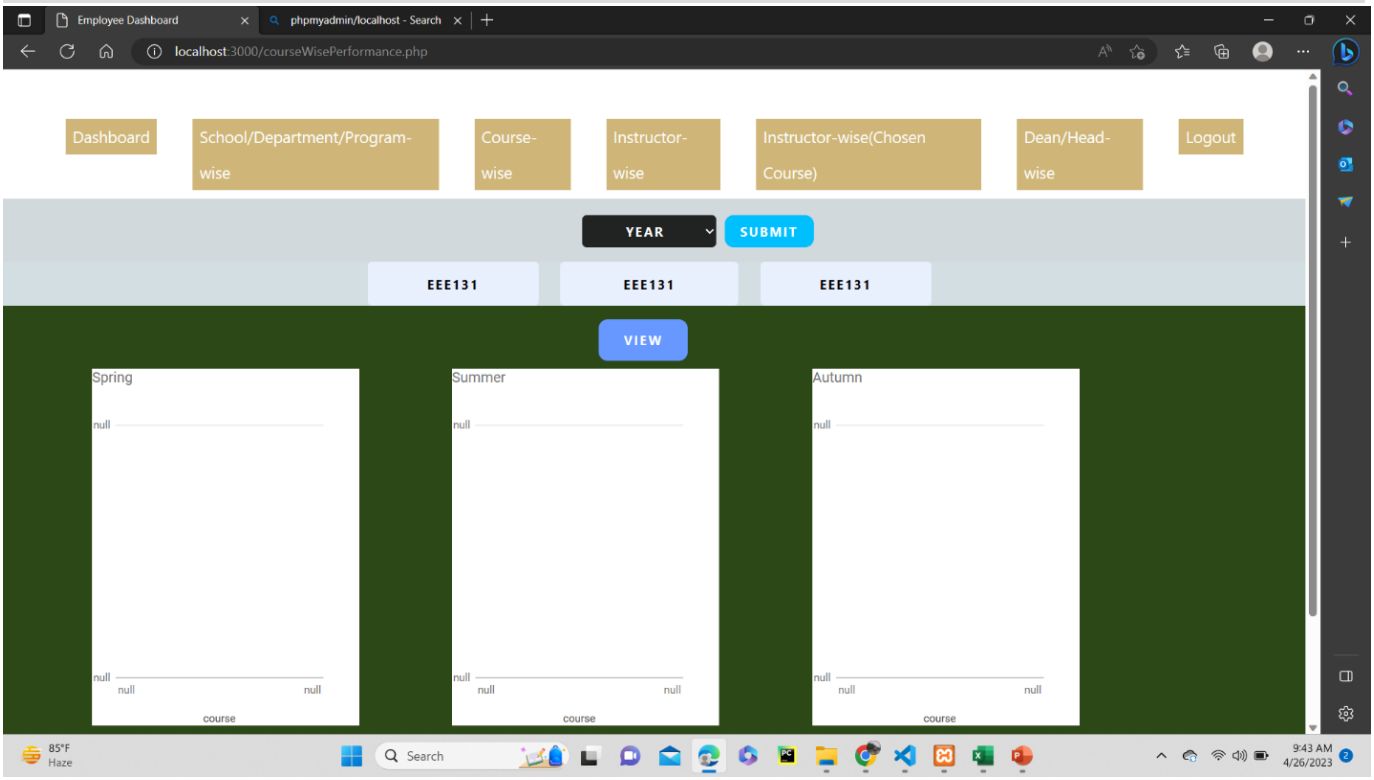

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/spiderChart.php
Dashboard
PLO Analysis
PLO Achievement Stats
Spider Chart Analysis
Data Entry
Enrollment Stats
GPA Analysis
Logout
ENTER STUDENT ID
ENTER
VIEW PO ANALYSIS
VIEW CO ANALYSIS
PO Achieved
PO 2
PO 3
PO 4
PO 5
PO 7
PO 9
PO 12
82
81
80
79
78
77
76
75
85°F
Haze
Search
9:42 AM
4/26/2023

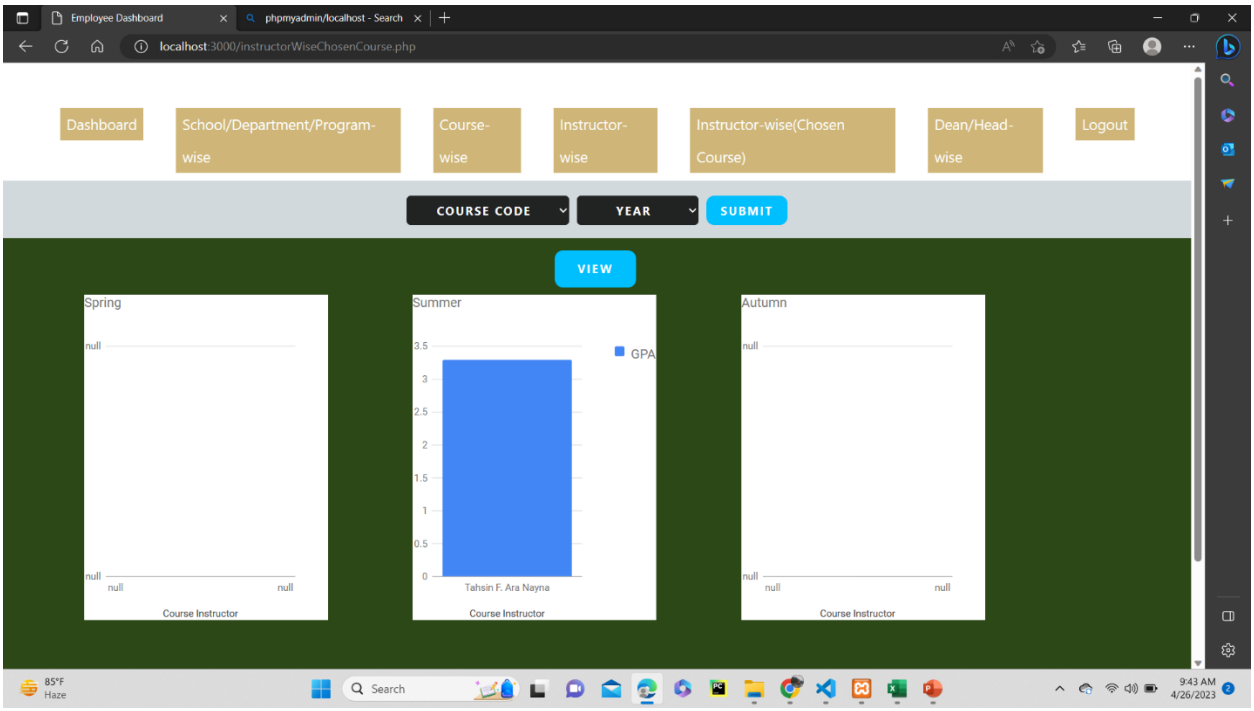

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/spiderChart.php
Dashboard
PLO Analysis
PLO Achievement Stats
Spider Chart Analysis
Data Entry
Enrollment Stats
GPA Analysis
Logout
ENTER STUDENT ID
ENTER
VIEW PO ANALYSIS
VIEW CO ANALYSIS
CO Achieved
CO 3
82
81
80
79
78
77
76
75
85°F
Haze
Search
9:42 AM
4/26/2023

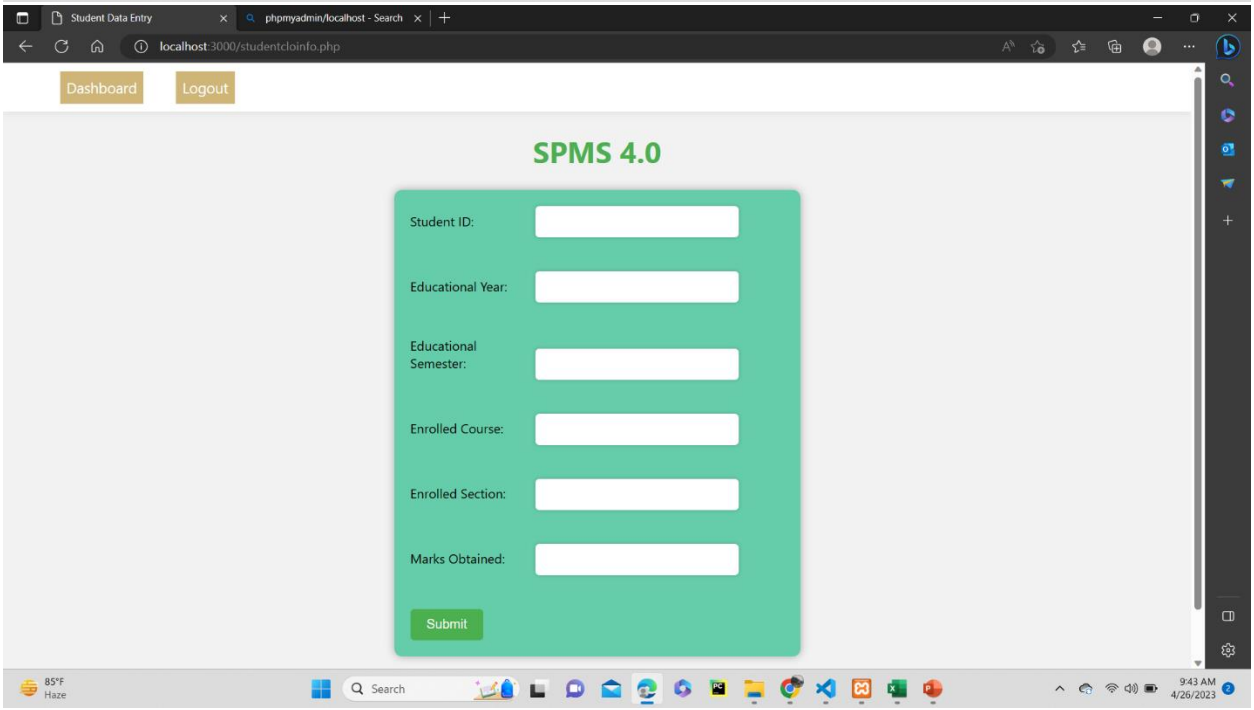

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/school_department_program_stats.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
YEAR
SUBMIT
SCHOOL-WISE GPA TREND
DEPARTMENT-WISE GPA TREND
PROGRAM-WISE GPA TREND
Spring
Summer
Autumn
GPA
SBE
SETS
SLASS
SETS
GPA
3.3
School
Search
85°F
Haze
9:43 AM
4/26/2023

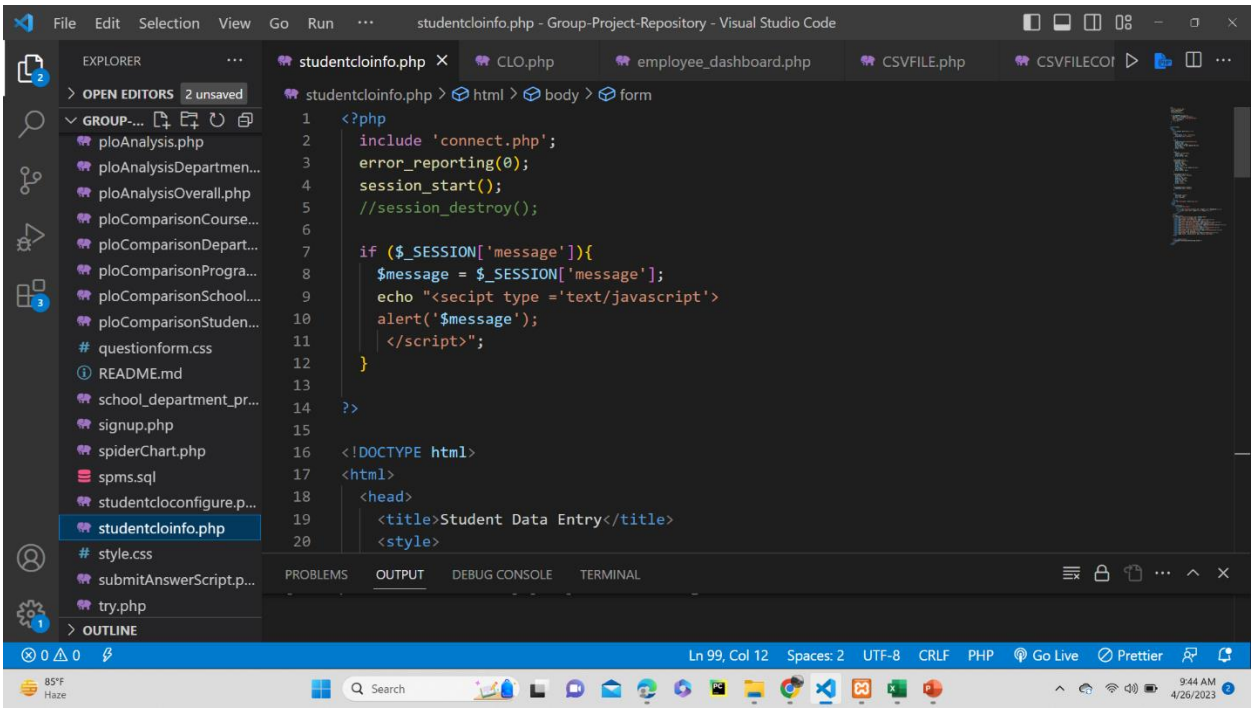

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/school_department_program_stats.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
YEAR
SUBMIT
SCHOOL-WISE GPA TREND
DEPARTMENT-WISE GPA TREND
PROGRAM-WISE GPA TREND
Spring
Summer
Autumn
GPA
CSE
GSG
HRM
ACN
EEE
MIS
FIN
Department
Search
85°F
Haze
9:43 AM
4/26/2023

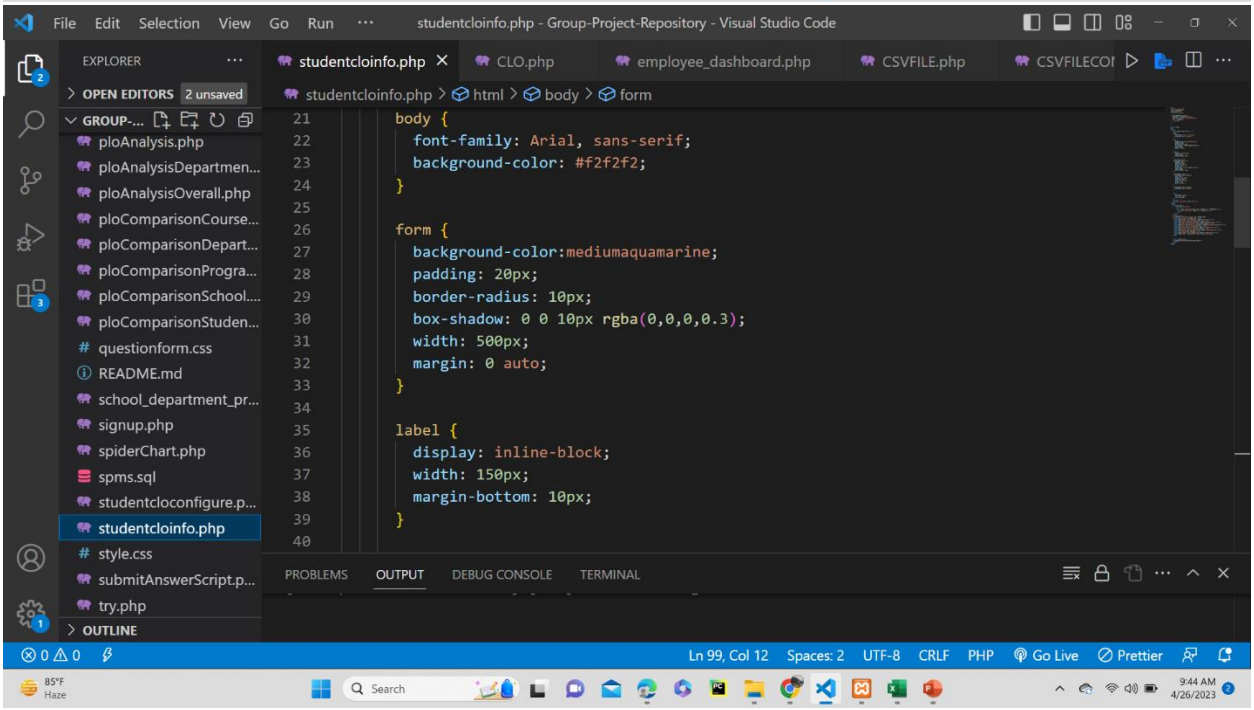

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/school_department_program_stats.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
YEAR
SUBMIT
SCHOOL-WISE GPA TREND
DEPARTMENT-WISE GPA TREND
PROGRAM-WISE GPA TREND
Spring
Summer
Autumn
GPA
BBA in HRM
BA in GSS
BBA in ACN
BSc in EEE
BBA in FIN
Program
85°F
Haze
Search
9:43 AM
4/26/2023

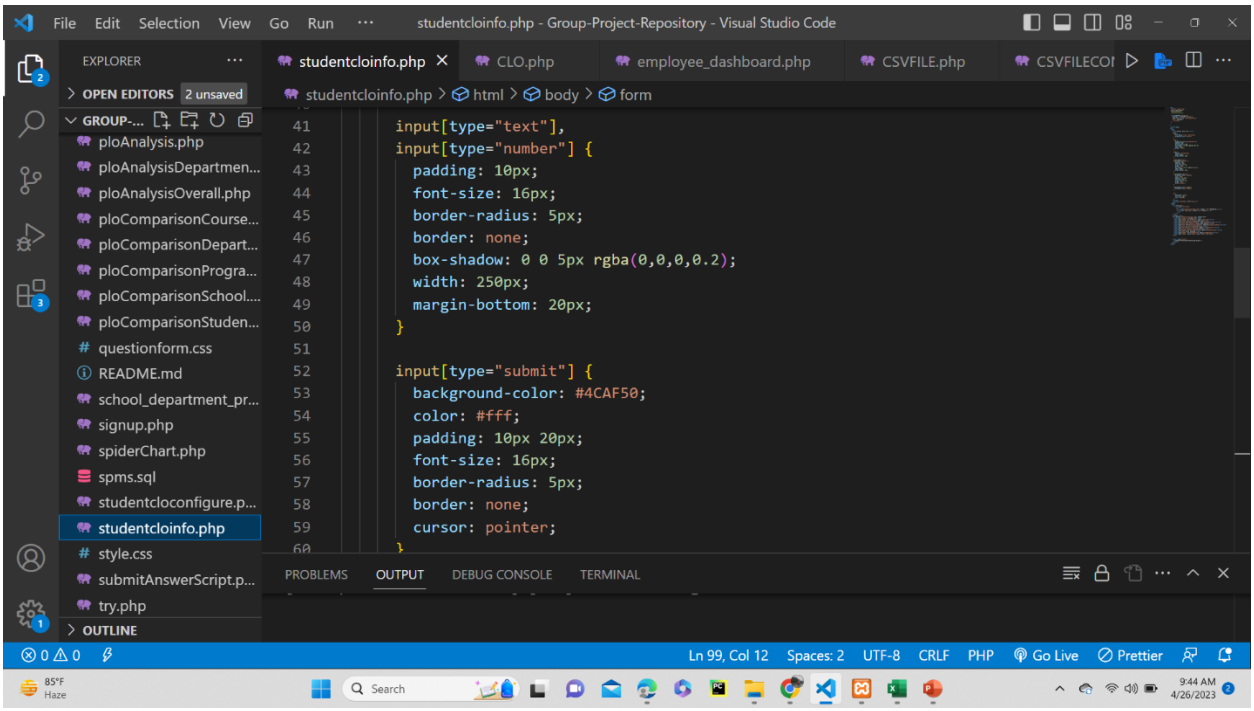

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/courseWisePerformance.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
YEAR
SUBMIT
EEE131
EEE131
EEE131
VIEW
Spring
Summer
Autumn
null
course
85°F
Haze
Search
9:43 AM
4/26/2023

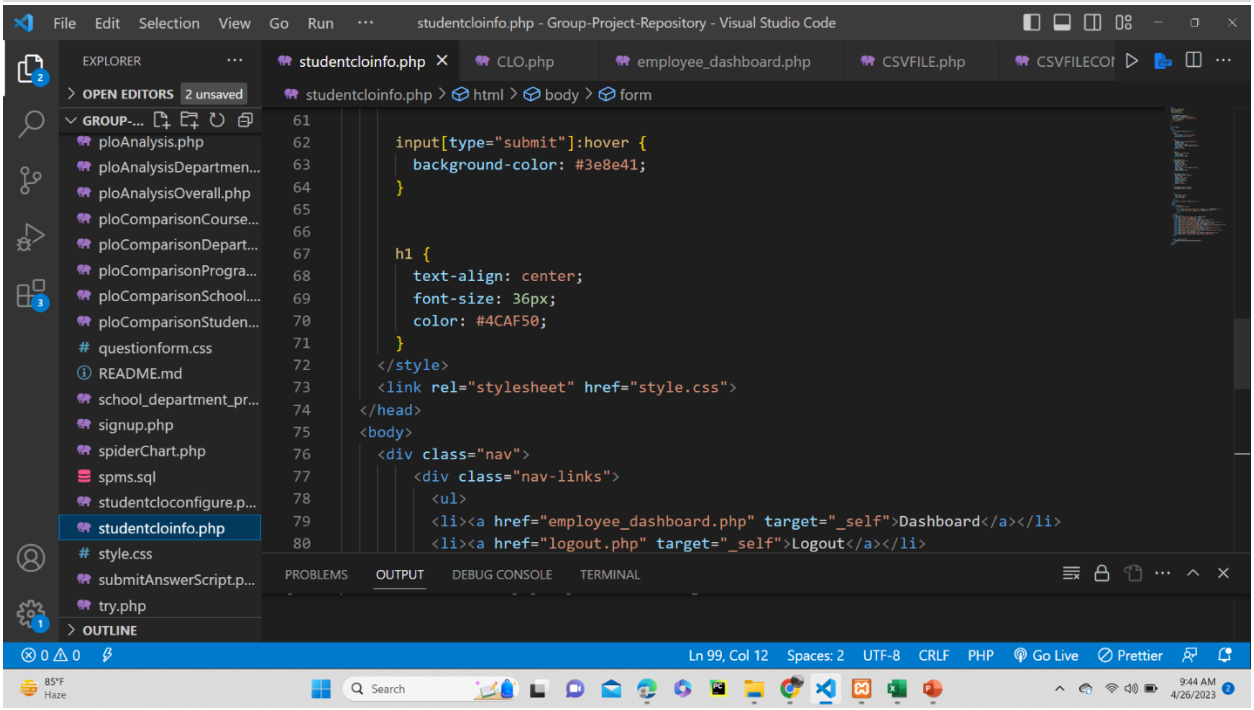

Employee Dashboard
phpmyadmin/localhost - Search
localhost:3000/instructorWiseChosenCourse.php
Dashboard
School/Department/Program-wise
Course-wise
Instructor-wise
Instructor-wise(Chosen Course)
Dean/Head-wise
Logout
COURSE CODE
YEAR
SUBMIT
VIEW
Spring
Summer
Autumn
GPA
Tahsin F. Ara Nayna
Course Instructor


Student Data Entry
phpmyadmin/localhost - Search
localhost:3000/studentcloinfo.php
Dashboard
Logout
SPMS 4.0
Student ID:
Educational Year:
Educational Semester:
Enrolled Course:
Enrolled Section:
Marks Obtained:
Submit

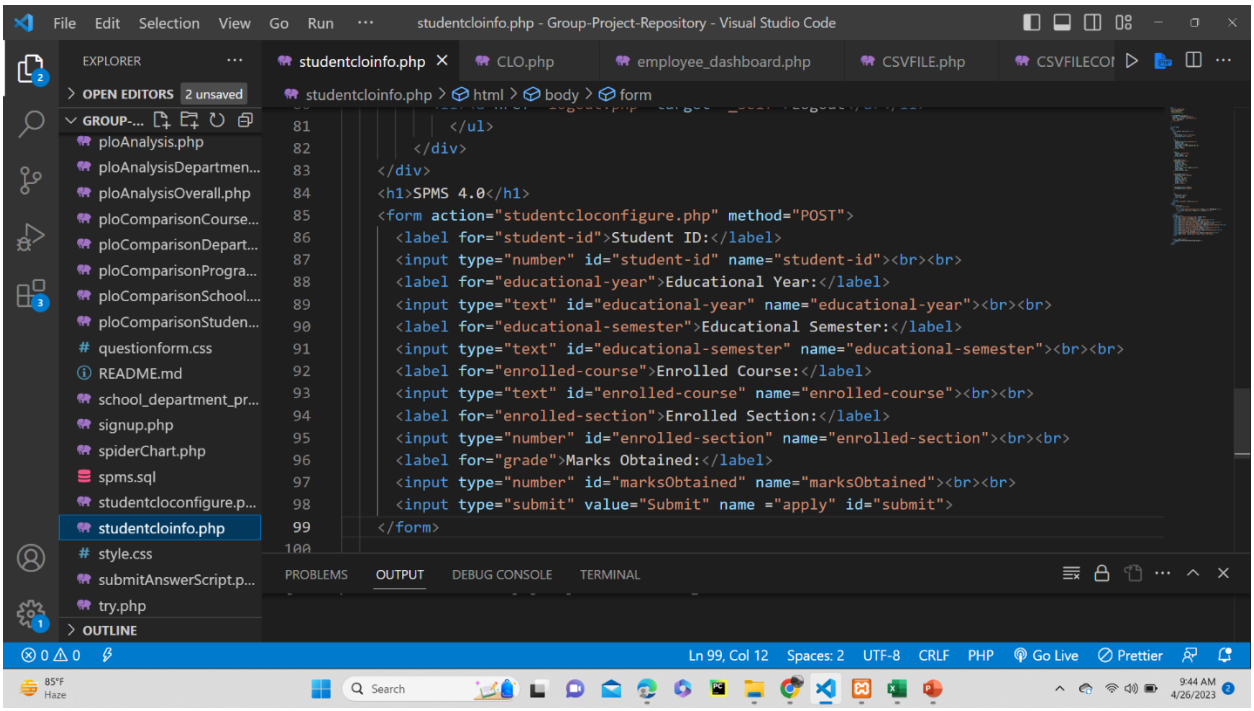

File Edit Selection View Go Run studentcloinfo.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
studentcloinfo.php > html > body > form
<?php
include 'connect.php';
error_reporting(0);
session_start();
//session_destroy();
if ($_SESSION['message']){
$message = $_SESSION['message'];
echo "<secipt type ='text/javascript'>
alert('$message');
</script>";
}
?>
<!DOCTYPE html>
<html>
<head>
<title>Student Data Entry</title>
<style>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 99, Col 12 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
Search


File Edit Selection View Go Run studentcloinfo.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
studentcloinfo.php > html > body > form
body {
font-family: Arial, sans-serif;
background-color: #f2f2f2;
}
form {
background-color:mediumaquamarine;
padding: 20px;
border-radius: 10px;
box-shadow: 0 0 10px rgba(0,0,0,0.3);
width: 500px;
margin: 0 auto;
}
label {
display: inline-block;
width: 150px;
margin-bottom: 10px;
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 99, Col 12 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
Search


File Edit Selection View Go Run studentcloinfo.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
studentcloinfo.php > html > body > form
input[type="text"],
input[type="number"] {
padding: 10px;
font-size: 16px;
border-radius: 5px;
border: none;
box-shadow: 0 0 5px rgba(0,0,0,0.2);
width: 250px;
margin-bottom: 20px;
}
input[type="submit"] {
background-color: #4CAF50;
color: #fff;
padding: 10px 20px;
font-size: 16px;
border-radius: 5px;
border: none;
cursor: pointer;
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 99, Col 12 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
Search
9:44 AM 4/26/2023


File Edit Selection View Go Run studentcloinfo.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
studentcloinfo.php > html > body > form
input[type="submit"]:hover {
background-color: #3e8e41;
}
h1 {
text-align: center;
font-size: 36px;
color: #4CAF50;
}
</style>
<link rel="stylesheet" href="style.css">
</head>
<body>

Dashboard
Logout
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 99, Col 12 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier
Search
9:44 AM 4/26/2023


File Edit Selection View Go Run studentcloinfo.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
OPEN EDITORS 2 unsaved
studentcloinfo.php > html > body > form
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE

<h1>SPMS 4.0</h1>
<form action="studentcloconfigure.php" method="POST">
<label for="student-id">Student ID:</label>
<input type="number" id="student-id" name="student-id">

<label for="educational-year">Educational Year:</label>
<input type="text" id="educational-year" name="educational-year">

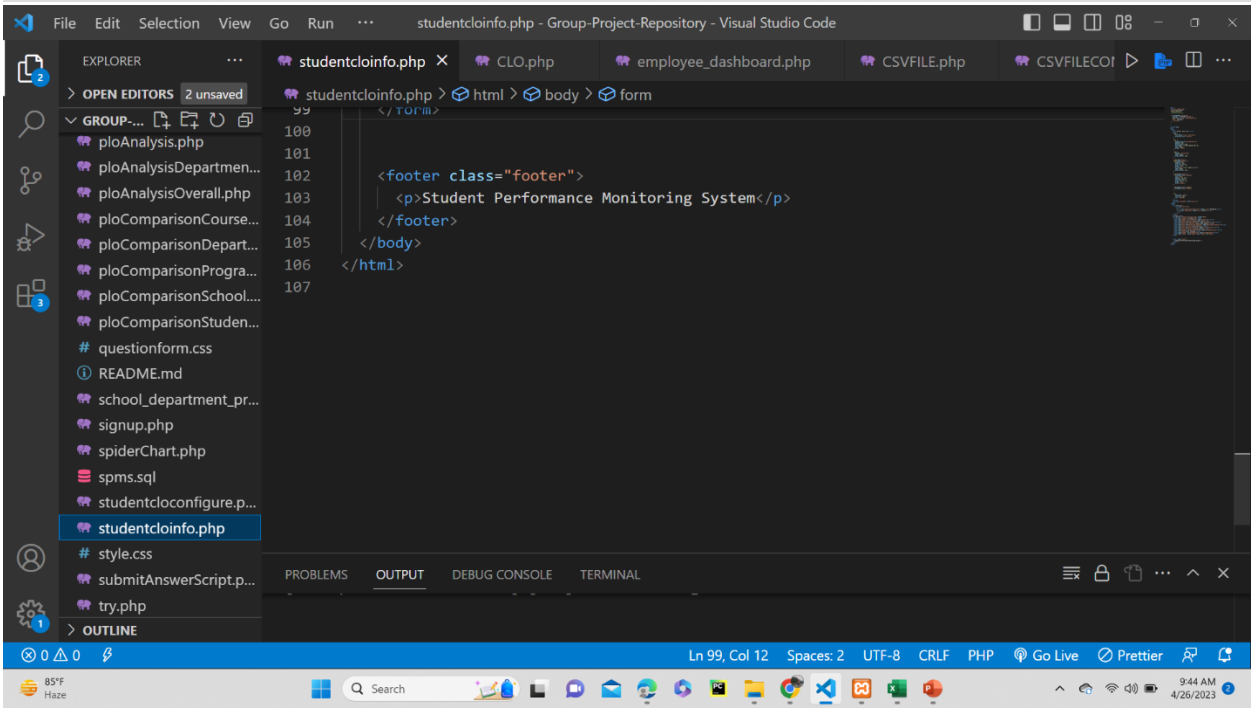
<label for="educational-semester">Educational Semester:</label>
<input type="text" id="educational-semester" name="educational-semester">

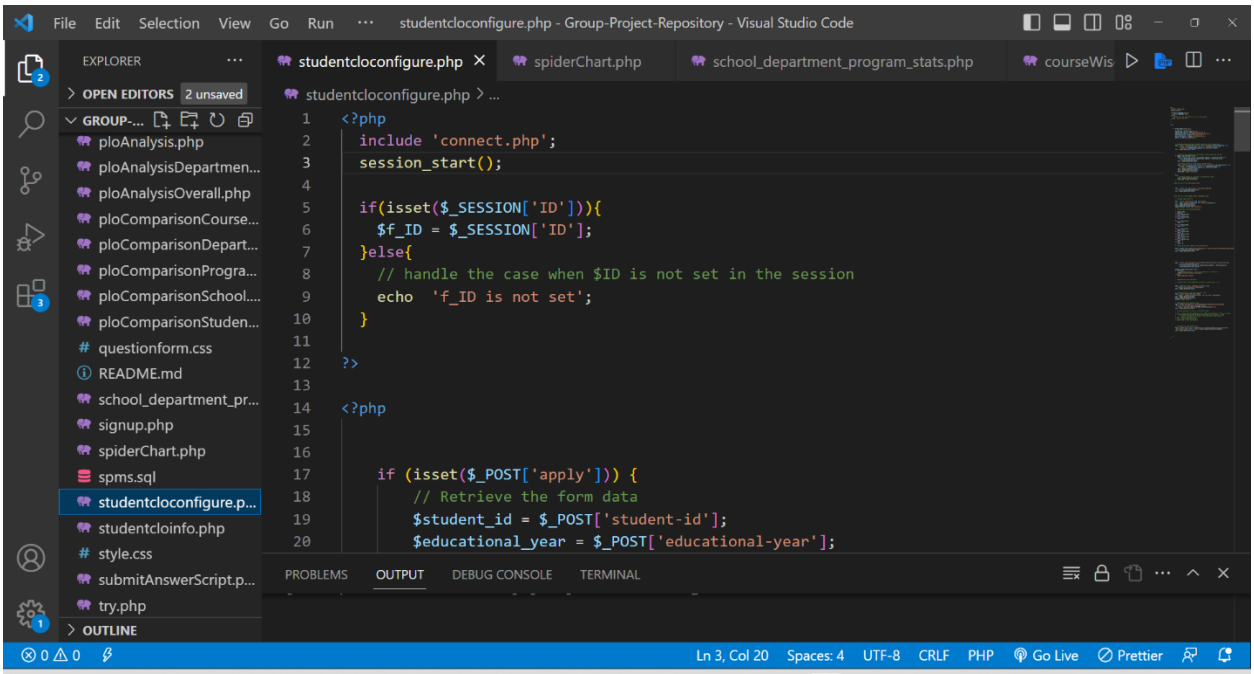
<label for="enrolled-course">Enrolled Course:</label>
<input type="text" id="enrolled-course" name="enrolled-course">

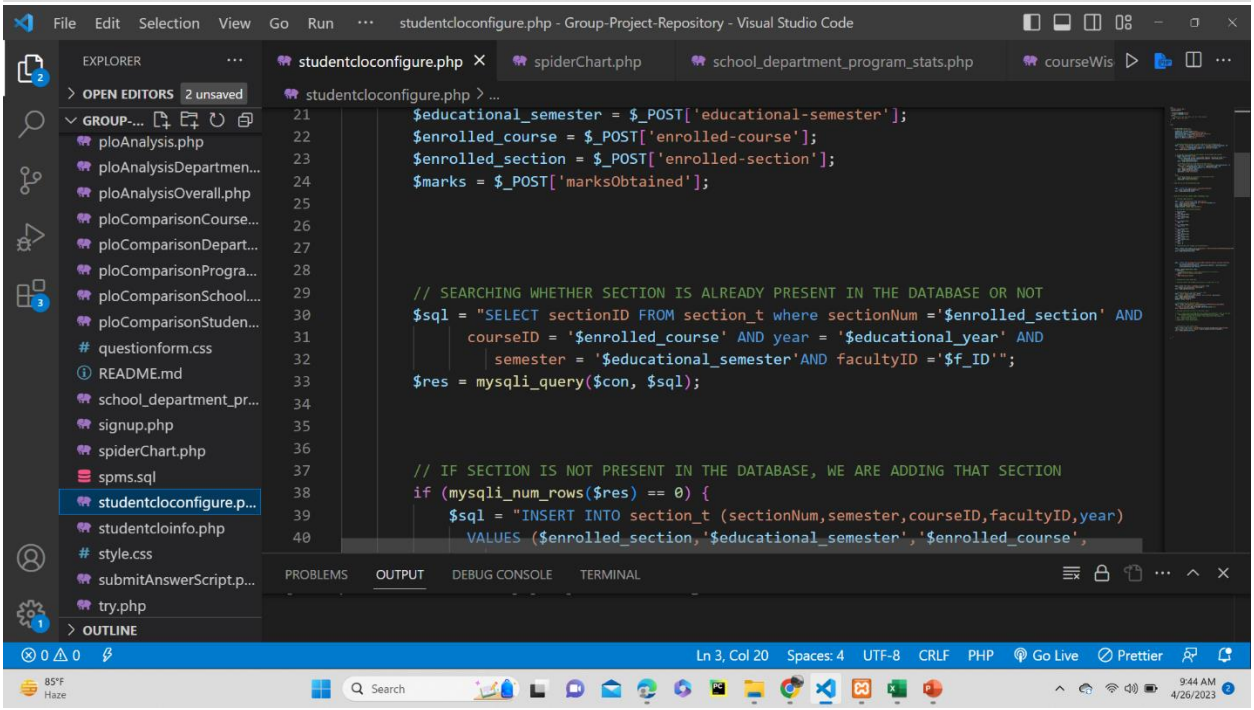
<label for="enrolled-section">Enrolled Section:</label>
<input type="number" id="enrolled-section" name="enrolled-section">

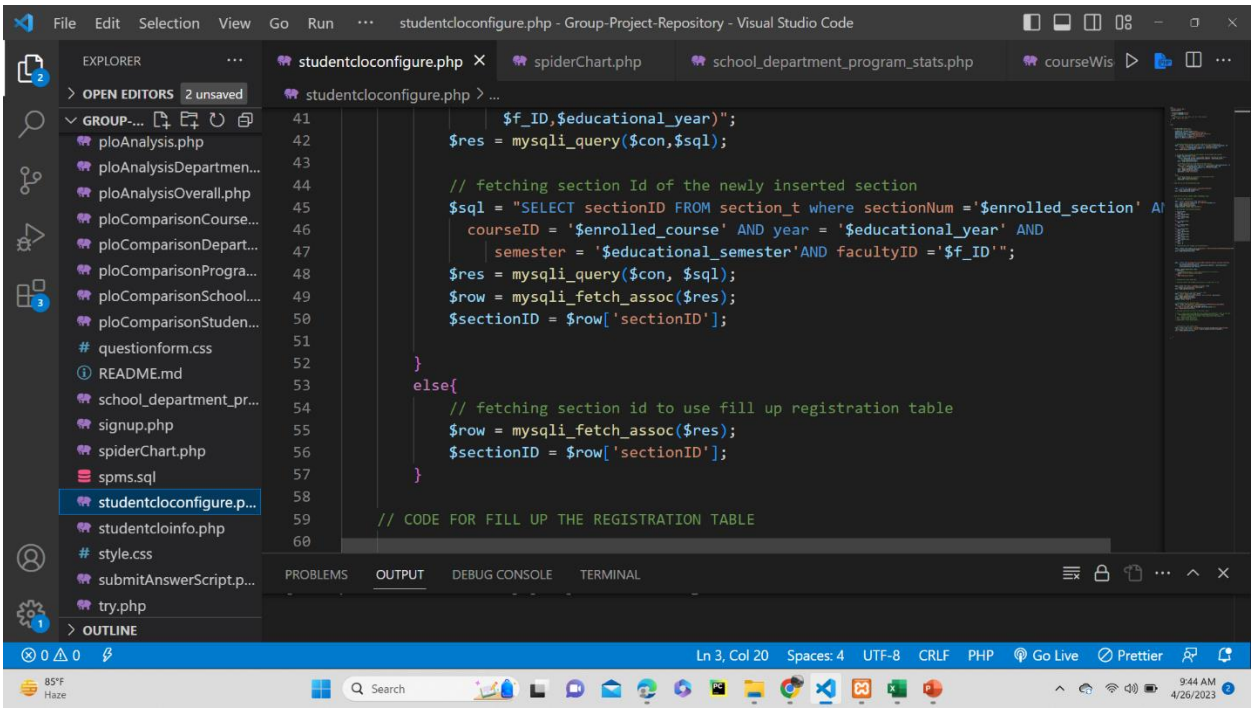
<label for="grade">Marks Obtained:</label>
<input type="number" id="marksObtained" name="marksObtained">

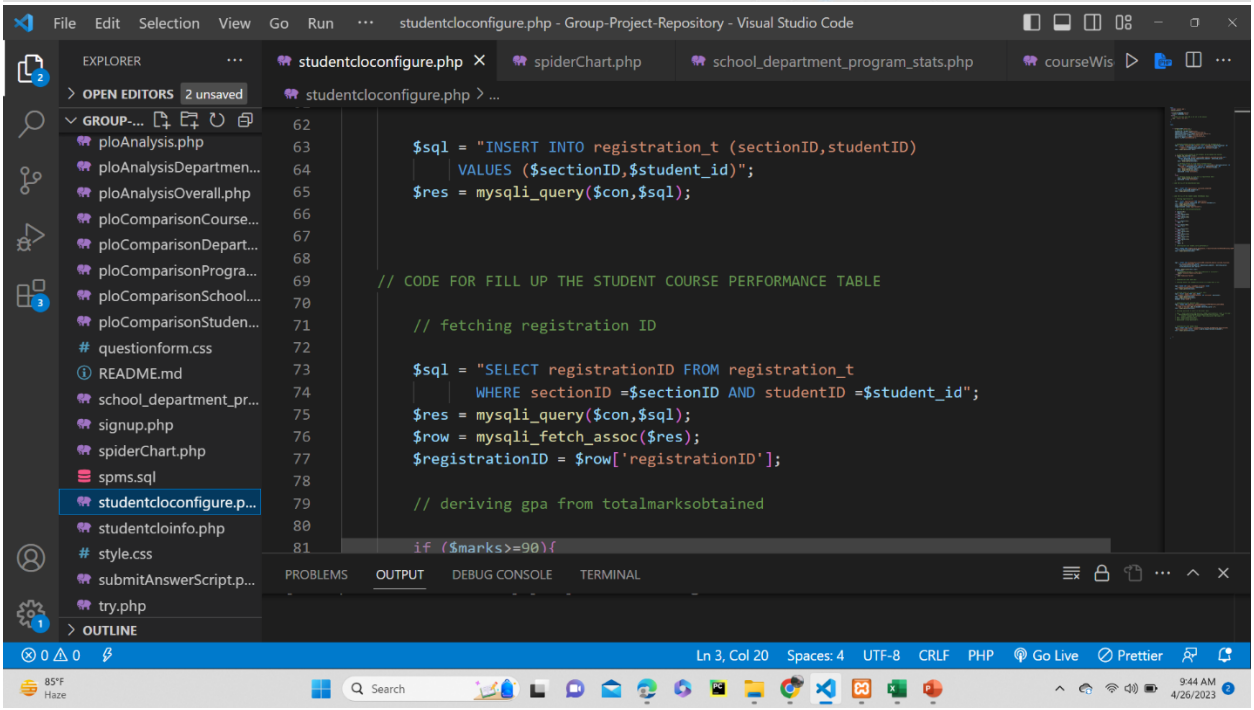
<input type="submit" value="Submit" name ="apply" id="submit">
</form>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 99, Col 12 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier

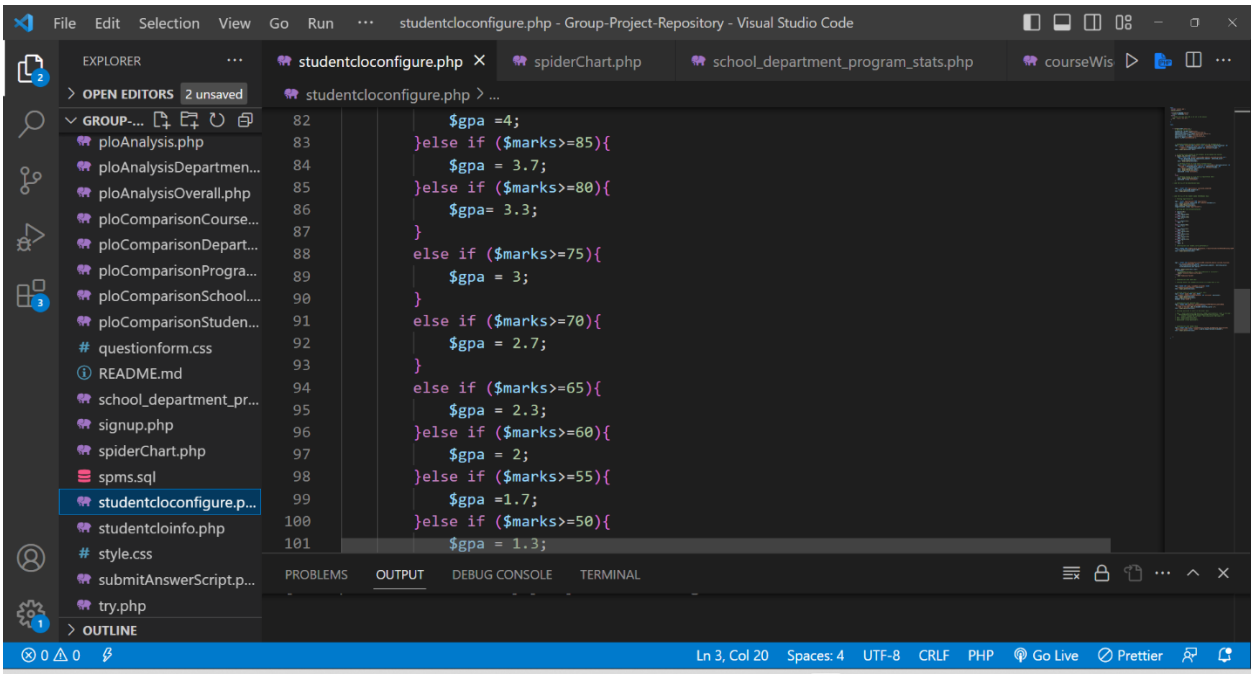

File Edit Selection View Go Run studentcloinfo.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
OPEN EDITORS 2 unsaved
studentcloinfo.php > html > body > form
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
<footer class="footer">
Student Performance Monitoring System
</footer>
</body>
</html>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 99, Col 12 Spaces: 2 UTF-8 CRLF PHP Go Live Prettier

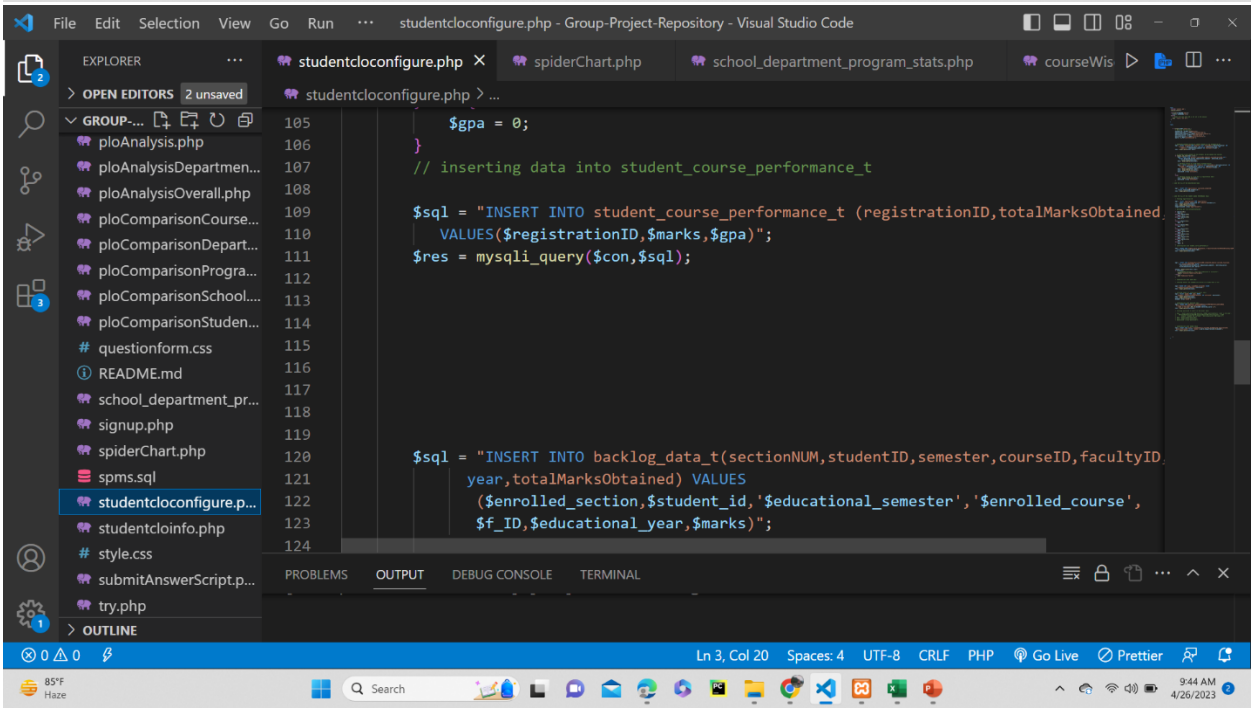

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
<?php
include 'connect.php';
session_start();
if(isset($_SESSION['ID'])){
$f_ID = $_SESSION['ID'];
}else{
// handle the case when $ID is not set in the session
echo 'f_ID is not set';
}
?>
<?php
if (isset($_POST['apply'])) {
// Retrieve the form data
$student_id = $_POST['student-id'];
$educational_year = $_POST['educational-year'];
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:44 AM 4/26/2023

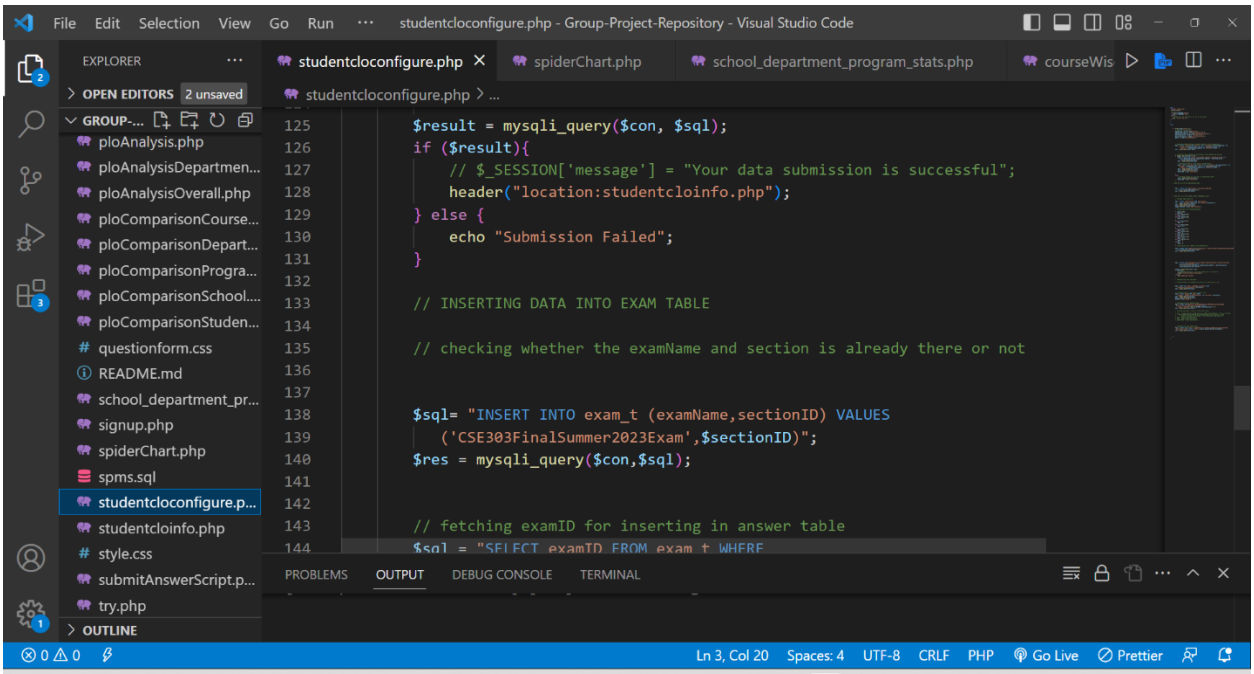

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
$educational_semester = $_POST['educational-semester'];
$enrolled_course = $_POST['enrolled-course'];
$enrolled_section = $_POST['enrolled-section'];
$marks = $_POST['marksObtained'];
// SEARCHING WHETHER SECTION IS ALREADY PRESENT IN THE DATABASE OR NOT
$sql = "SELECT sectionID FROM section_t where sectionNum ='$enrolled_section' AND
courseID = '$enrolled_course' AND year = '$educational_year' AND
semester = '$educational_semester'AND facultyID ='$f_ID'";
$res = mysqli_query($con, $sql);
// IF SECTION IS NOT PRESENT IN THE DATABASE, WE ARE ADDING THAT SECTION
if (mysqli_num_rows($res) == 0) {
$sql = "INSERT INTO section_t (sectionNum,semester,courseID,facultyID,year)
VALUES ($enrolled_section,'$educational_semester','$enrolled_course',
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:44 AM 4/26/2023

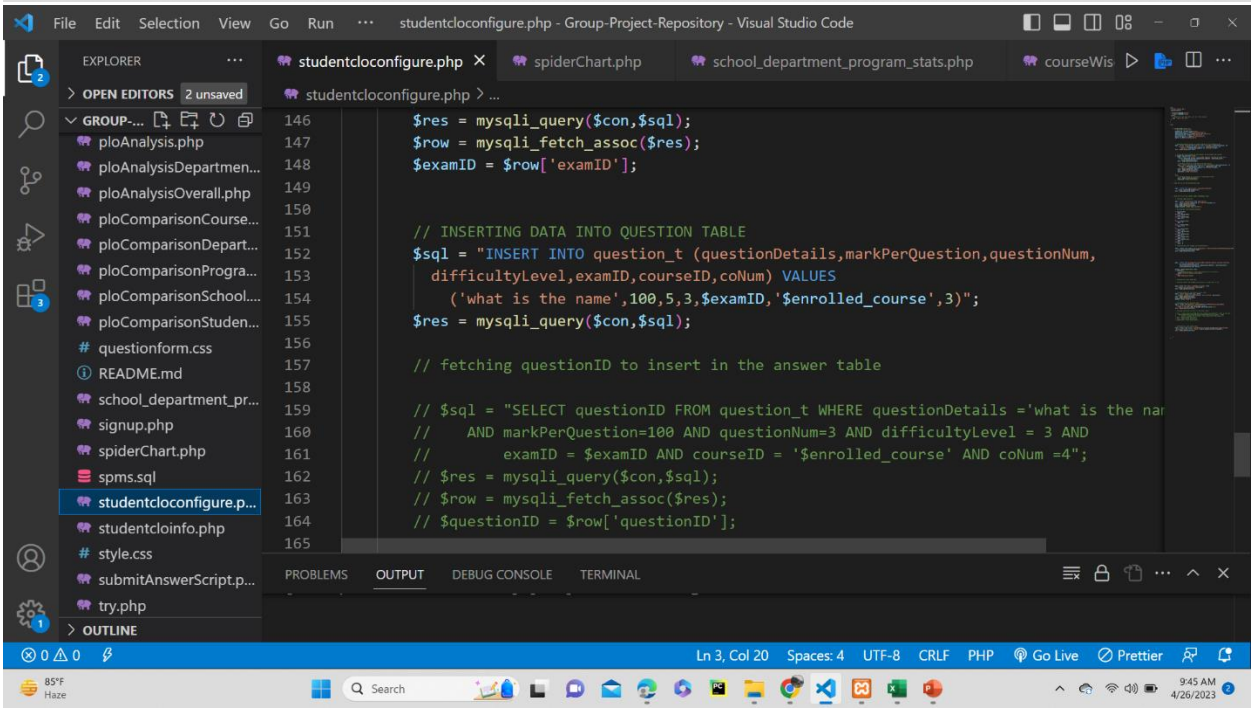

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
$f_ID,$educational_year)";
$res = mysqli_query($con,$sql);
// fetching section Id of the newly inserted section
$sql = "SELECT sectionID FROM section_t where sectionNum ='$enrolled_section' A
courseID = '$enrolled_course' AND year = '$educational_year' AND
semester = '$educational_semester'AND facultyID ='$f_ID'";
$res = mysqli_query($con, $sql);
$row = mysqli_fetch_assoc($res);
$sectionID = $row['sectionID'];
}
else{
// fetching section id to use fill up registration table
$row = mysqli_fetch_assoc($res);
$sectionID = $row['sectionID'];
}
// CODE FOR FILL UP THE REGISTRATION TABLE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:44 AM 4/26/2023

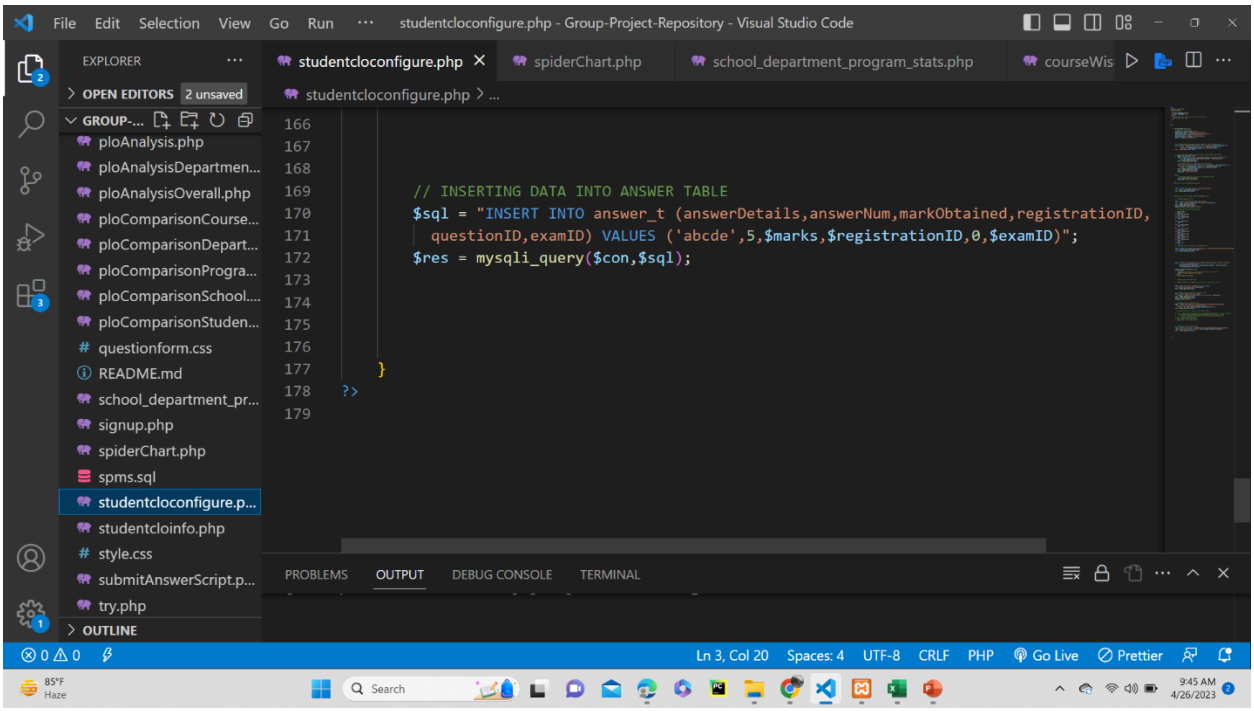

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
$sql = "INSERT INTO registration_t (sectionID,studentID)
VALUES ($sectionID,$student_id)";
$res = mysqli_query($con,$sql);
// CODE FOR FILL UP THE STUDENT COURSE PERFORMANCE TABLE
// fetching registration ID
$sql = "SELECT registrationID FROM registration_t
WHERE sectionID =$sectionID AND studentID =$student_id";
$res = mysqli_query($con,$sql);
$row = mysqli_fetch_assoc($res);
$registrationID = $row['registrationID'];
// deriving gpa from totalmarksobtained
if ($marks>=90){
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:44 AM 4/26/2023

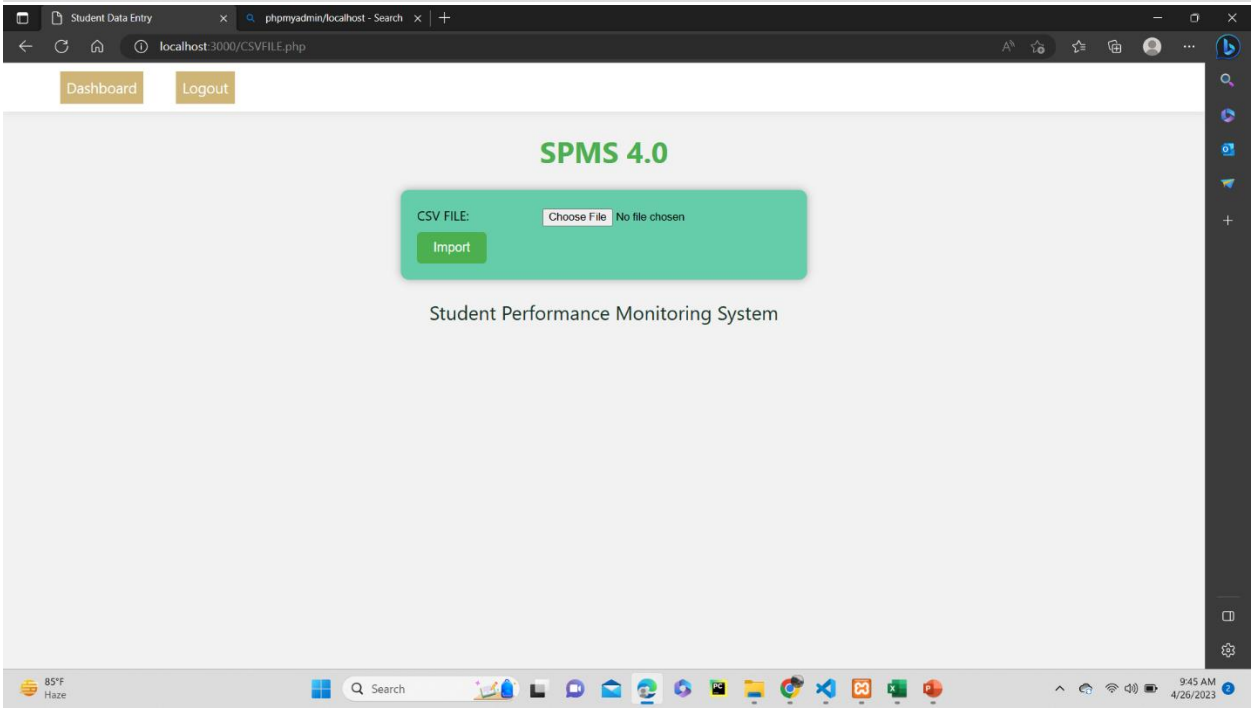

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
$gpa =4;
}else if ($marks>=85){
$gpa = 3.7;
}else if ($marks>=80){
$gpa= 3.3;
}
else if ($marks>=75){
$gpa = 3;
}
else if ($marks>=70){
$gpa = 2.7;
}
else if ($marks>=65){
$gpa = 2.3;
}else if ($marks>=60){
$gpa = 2;
}else if ($marks>=55){
$gpa =1.7;
}else if ($marks>=50){
$gpa = 1.3;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
Search
9:44 AM 4/26/2023

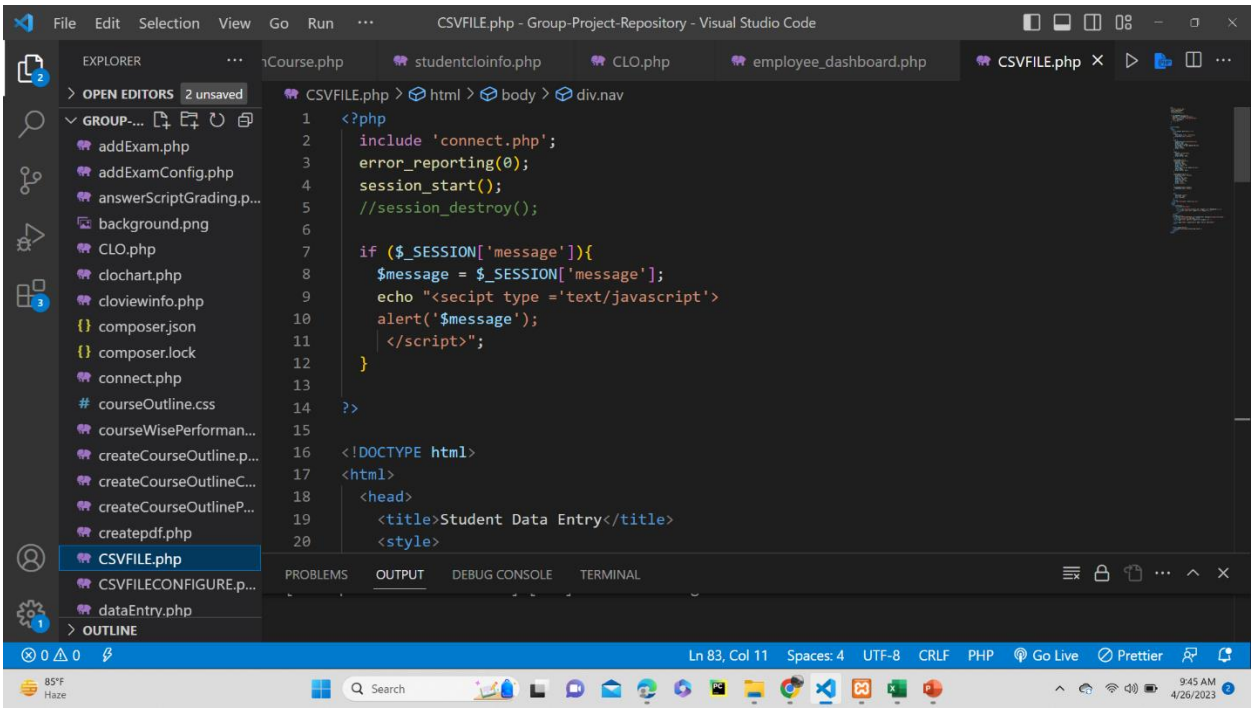

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
$gpa = 0;
}
// inserting data into student_course_performance_t
$sql = "INSERT INTO student_course_performance_t (registrationID,totalMarksObtained,
VALUES($registrationID,$marks,$gpa)";
$res = mysqli_query($con,$sql);
$sql = "INSERT INTO backlog_data_t(sectionNUM,studentID,semester,courseID,facultyID,
year,totalMarksObtained) VALUES
($enrolled_section,$student_id,'$educational_semester','$enrolled_course',
$f_ID,$educational_year,$marks)";
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
Search
9:44 AM 4/26/2023

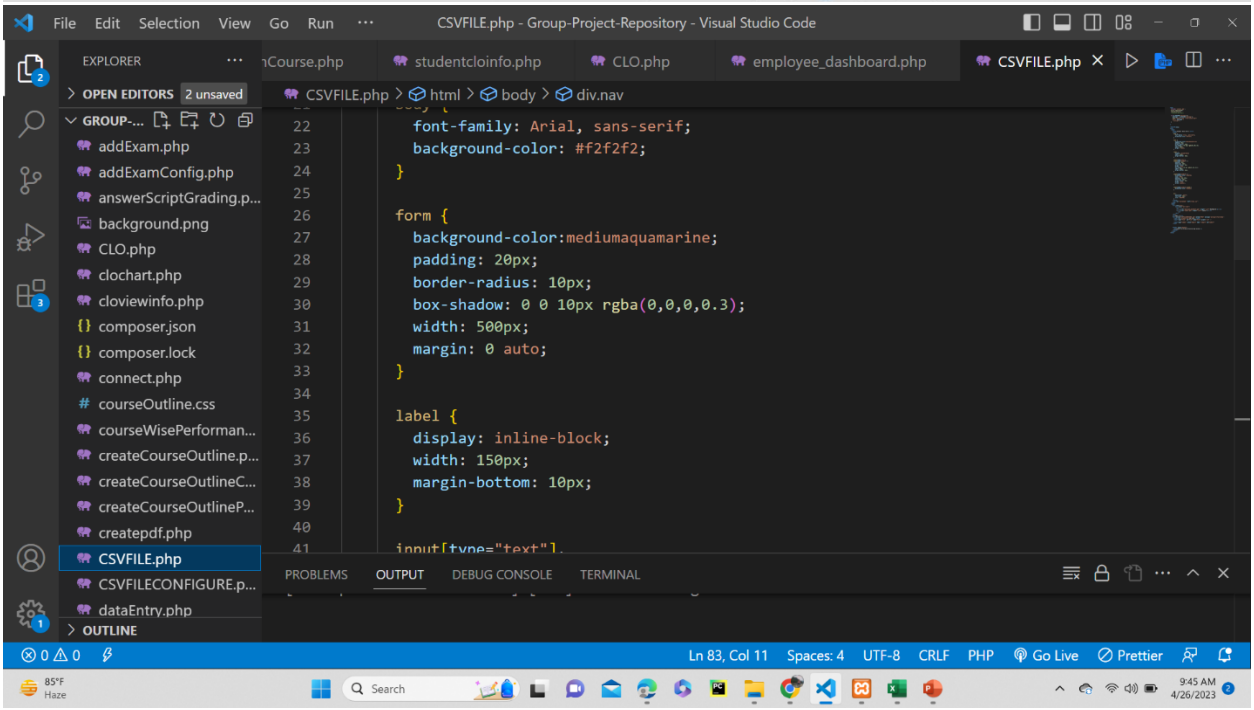

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
$result = mysqli_query($con, $sql);
if ($result){
// $_SESSION['message'] = "Your data submission is successful";
header("location:studentcloinfo.php");
} else {
echo "Submission Failed";
}
// INSERTING DATA INTO EXAM TABLE
// checking whether the examName and section is already there or not
$sql= "INSERT INTO exam_t (examName,sectionID) VALUES
('CSE303FinalSummer2023Exam',$sectionID)";
$res = mysqli_query($con,$sql);
// fetching examID for inserting in answer table
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier

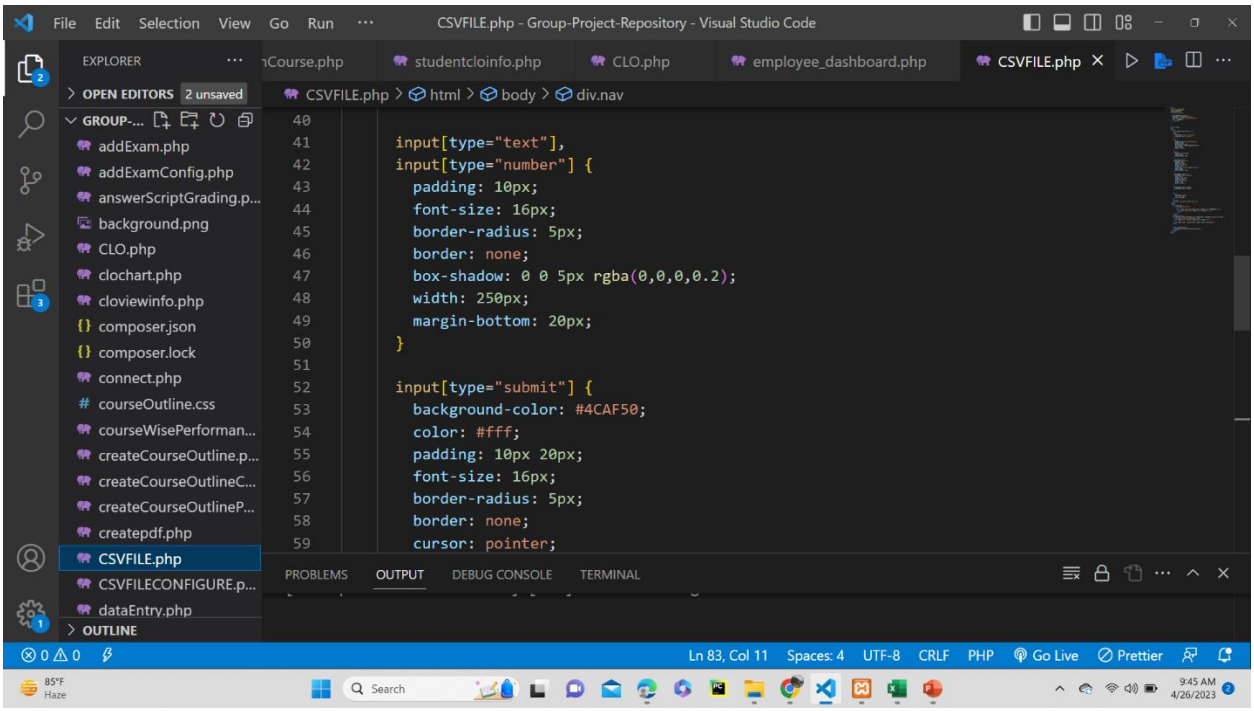

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
courseWis
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
$res = mysqli_query($con,$sql);
$row = mysqli_fetch_assoc($res);
$examID = $row['examID'];
// INSERTING DATA INTO QUESTION TABLE
$sql = "INSERT INTO question_t (questionDetails,markPerQuestion,questionNum,
difficultyLevel,examID,courseID,coNum) VALUES
('what is the name',100,5,3,$examID,'$enrolled_course',3)";
$res = mysqli_query($con,$sql);
// fetching questionID to insert in the answer table
// $sql = "SELECT questionID FROM question_t WHERE questionDetails ='what is the nar
// AND markPerQuestion=100 AND questionNum=3 AND difficultyLevel = 3 AND
// examID = $examID AND courseID = '$enrolled_course' AND coNum =4";
// $res = mysqli_query($con,$sql);
// $row = mysqli_fetch_assoc($res);
// $questionID = $row['questionID'];
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier

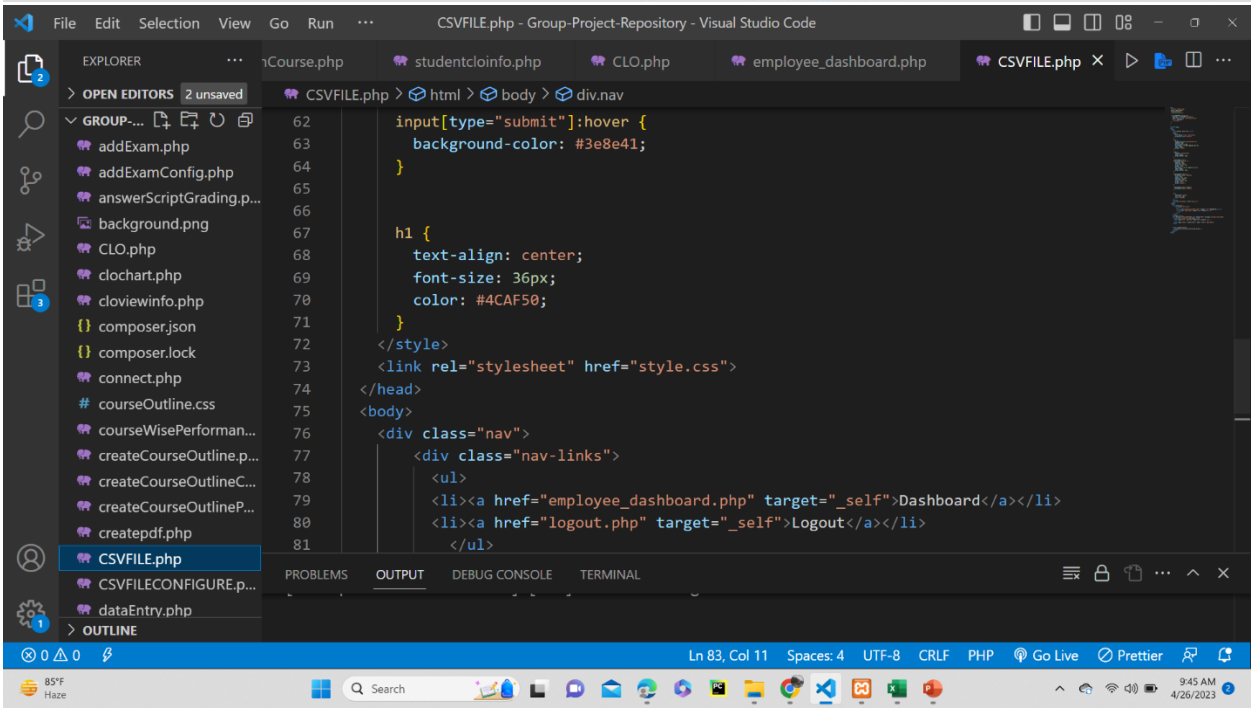

File Edit Selection View Go Run studentcloconfigure.php - Group-Project-Repository - Visual Studio Code
EXPLORER
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
OPEN EDITORS 2 unsaved
GROUP-...
ploAnalysis.php
ploAnalysisDepartmen...
ploAnalysisOverall.php
ploComparisonCourse...
ploComparisonDepart...
ploComparisonProgra...
ploComparisonSchool...
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
OUTLINE
166
167
168
169 // INSERTING DATA INTO ANSWER TABLE
170 $sql = "INSERT INTO answer_t (answerDetails,answerNum,markObtained,registrationID,
171 questionID,examID) VALUES ('abcde',5,$marks,$registrationID,0,$examID)";
172 $res = mysqli_query($con,$sql);
173
174
175
176
177 }
178 ?>
179
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 3, Col 20 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
Search
9:45 AM 4/26/2023


Student Data Entry
phpmyadmin/localhost - Search
localhost:3000/CSVFILE.php
Dashboard
Logout
SPMS 4.0
CSV FILE:
Choose File No file chosen
Import
Student Performance Monitoring System
Search
9:45 AM 4/26/2023


File Edit Selection View Go Run
CSVFILE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILE.php > html > body > div.nav
1 <?php
2 include 'connect.php';
3 error_reporting(0);
4 session_start();
5 //session_destroy();
6
7 if ($_SESSION['message']){
8 $message = $_SESSION['message'];
9 echo "<secipt type ='text/javascript'>
10 alert('$message');
11 </script>";
12 }
13
14 ?>
15
16 <!DOCTYPE html>
17 <html>
18 <head>
19 <title>Student Data Entry</title>
20 <style>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 83, Col 11 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

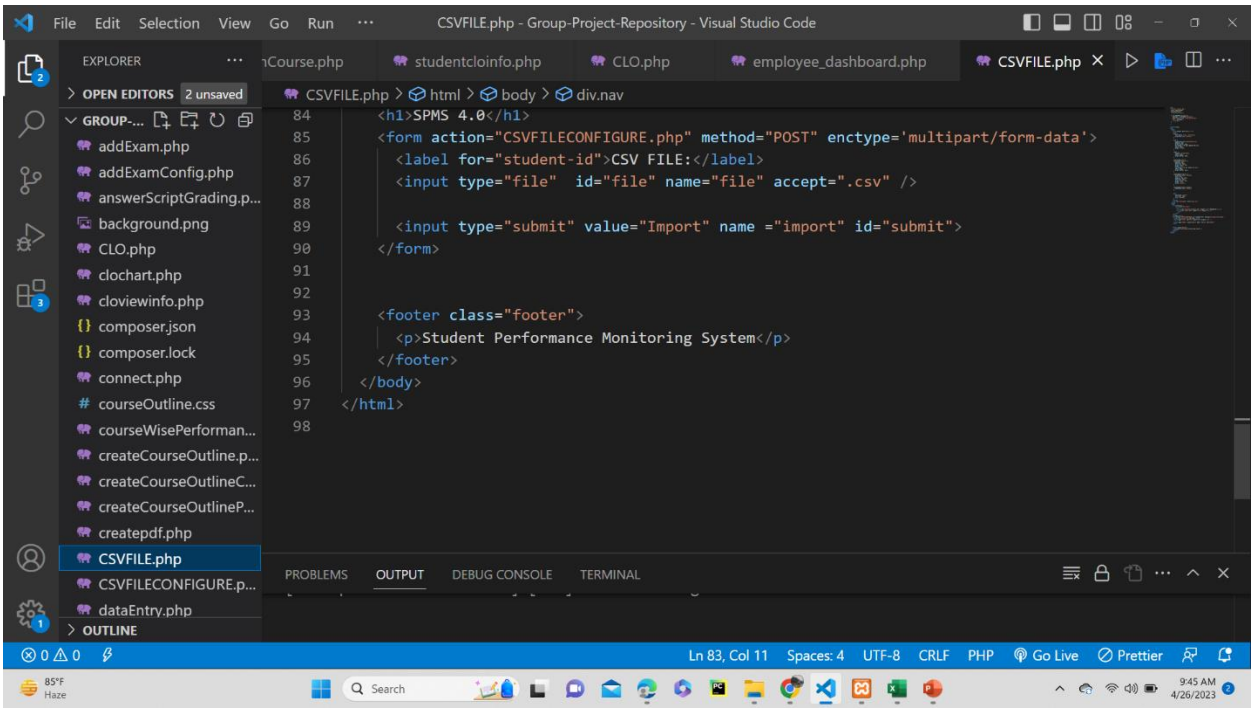

File Edit Selection View Go Run
CSVFILE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILE.php > html > body > div.nav
22 font-family: Arial, sans-serif;
23 background-color: #f2f2f2;
24 }
25
26 form {
27 background-color:mediumaquamarine;
28 padding: 20px;
29 border-radius: 10px;
30 box-shadow: 0 0 10px rgba(0,0,0,0.3);
31 width: 500px;
32 margin: 0 auto;
33 }
34
35 label {
36 display: inline-block;
37 width: 150px;
38 margin-bottom: 10px;
39 }
40
41 input[type="text"],
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 83, Col 11 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

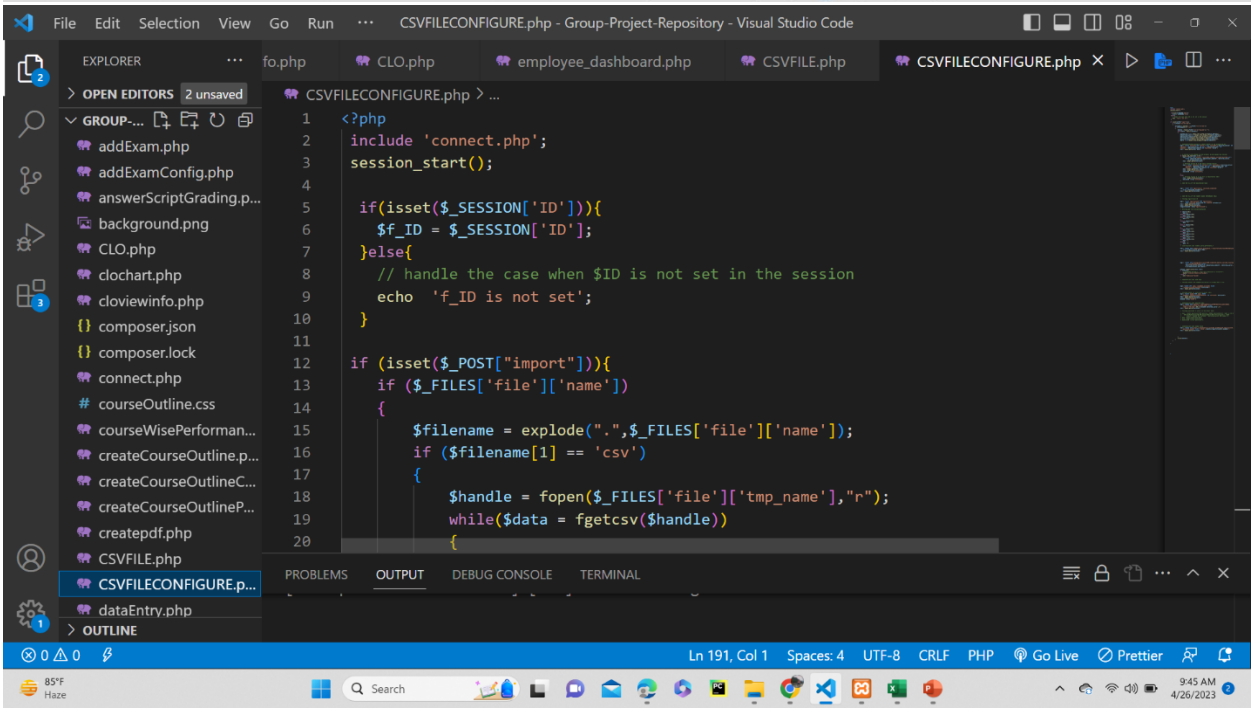

File Edit Selection View Go Run
CSVFILE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILE.php > html > body > div.nav
input[type="text"],
input[type="number"] {
padding: 10px;
font-size: 16px;
border-radius: 5px;
border: none;
box-shadow: 0 0 5px rgba(0,0,0,0.2);
width: 250px;
margin-bottom: 20px;
}
input[type="submit"] {
background-color: #4CAF50;
color: #fff;
padding: 10px 20px;
font-size: 16px;
border-radius: 5px;
border: none;
cursor: pointer;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 83, Col 11 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
Search
9:45 AM 4/26/2023

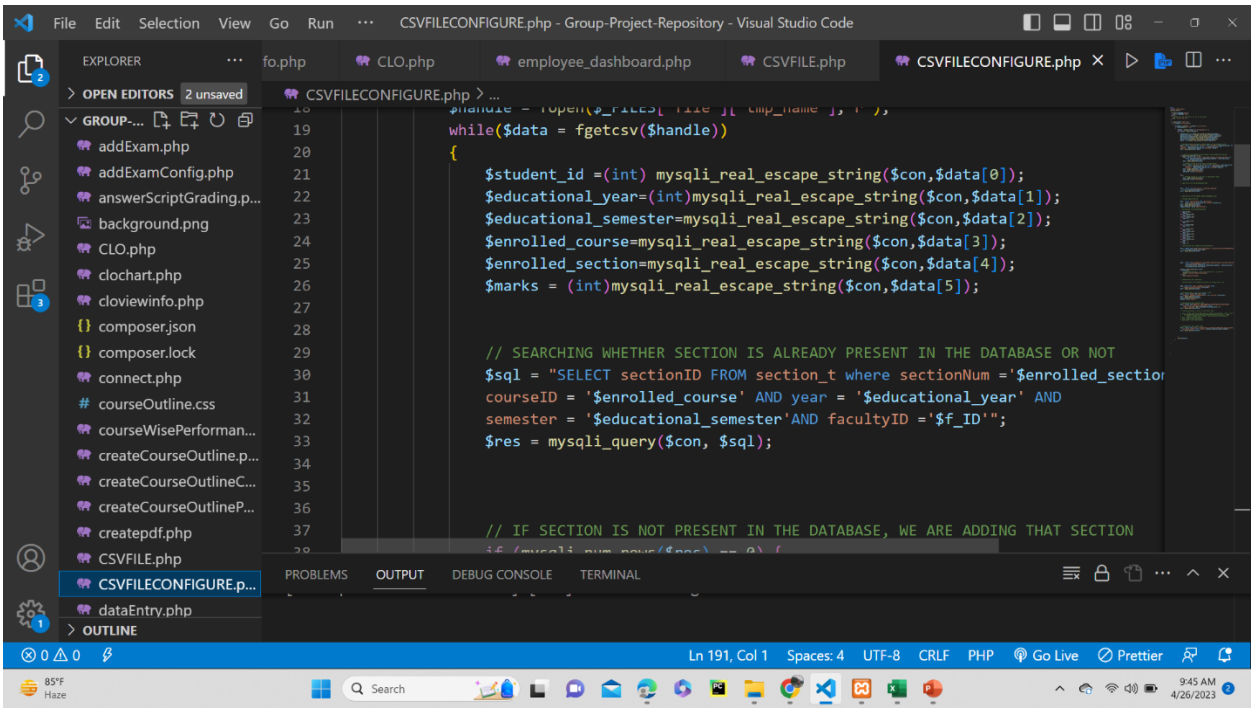

File Edit Selection View Go Run
CSVFILE.php - Group-Project-Repository - Visual Studio Code
input[type="submit"]:hover {
background-color: #3e8e41;
}
h1 {
text-align: center;
font-size: 36px;
color: #4CAF50;
}
</style>
<link rel="stylesheet" href="style.css">
</head>
<body>

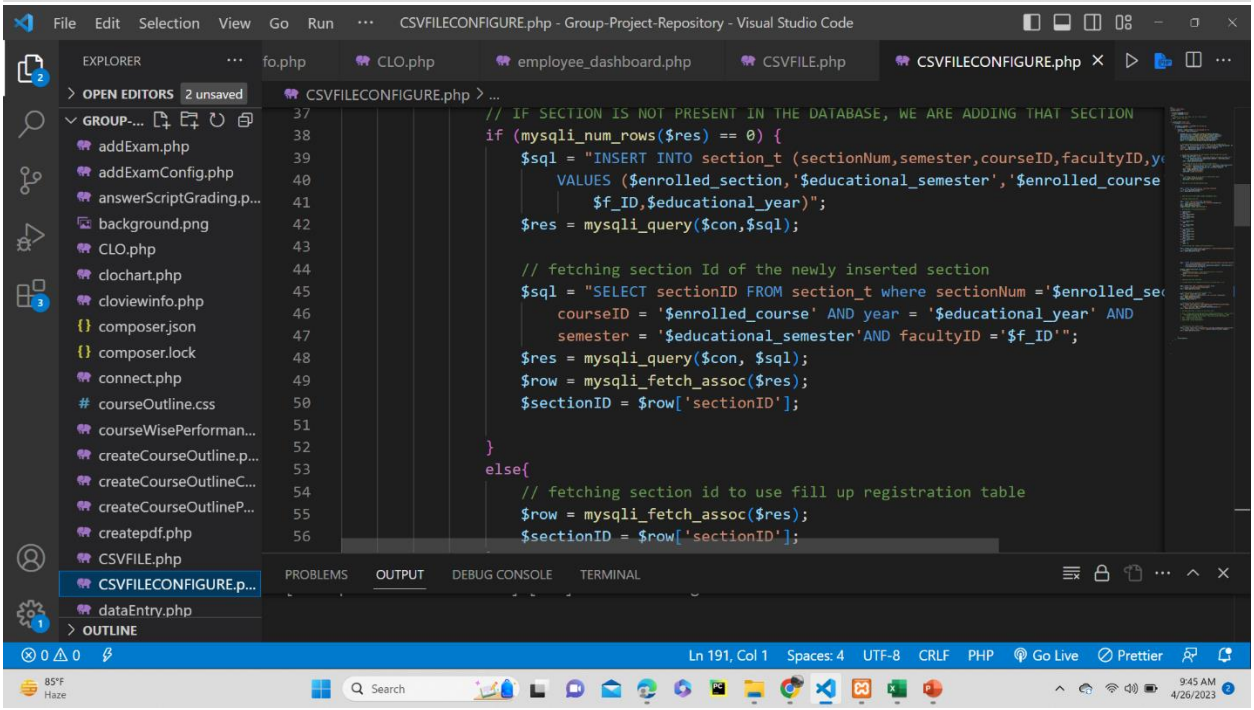
Dashboard
Logout

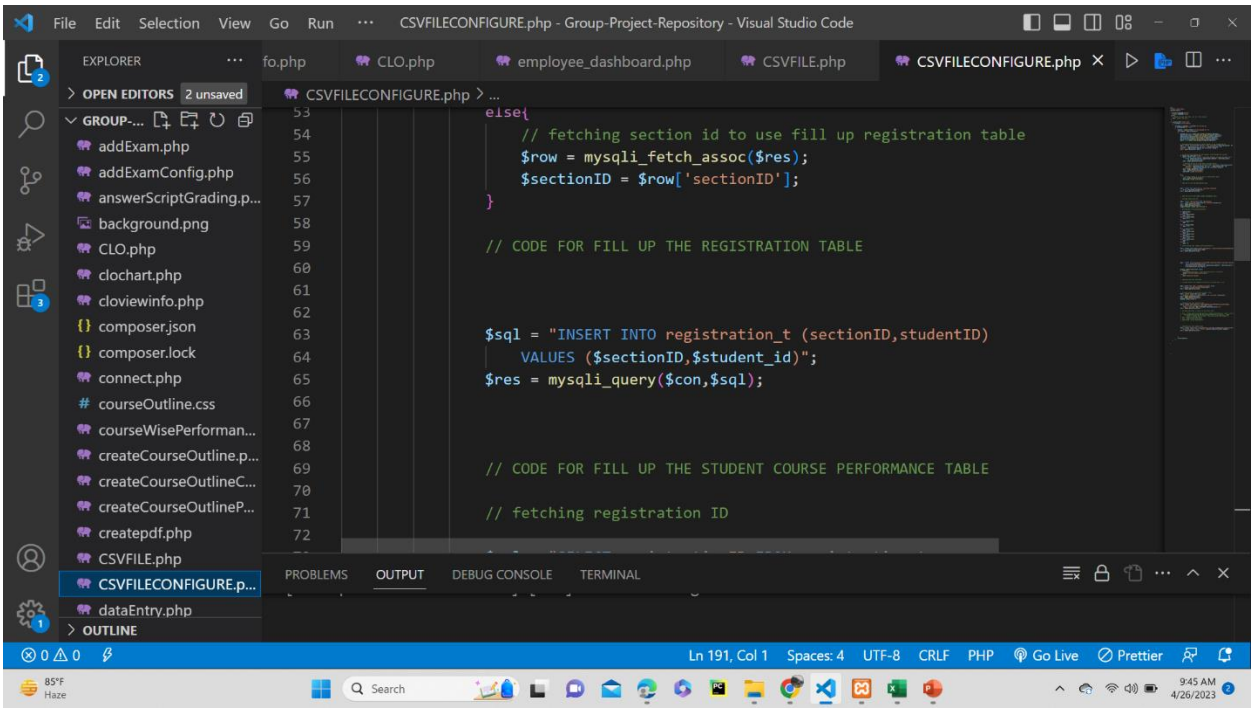
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 83, Col 11 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
Search
9:45 AM 4/26/2023

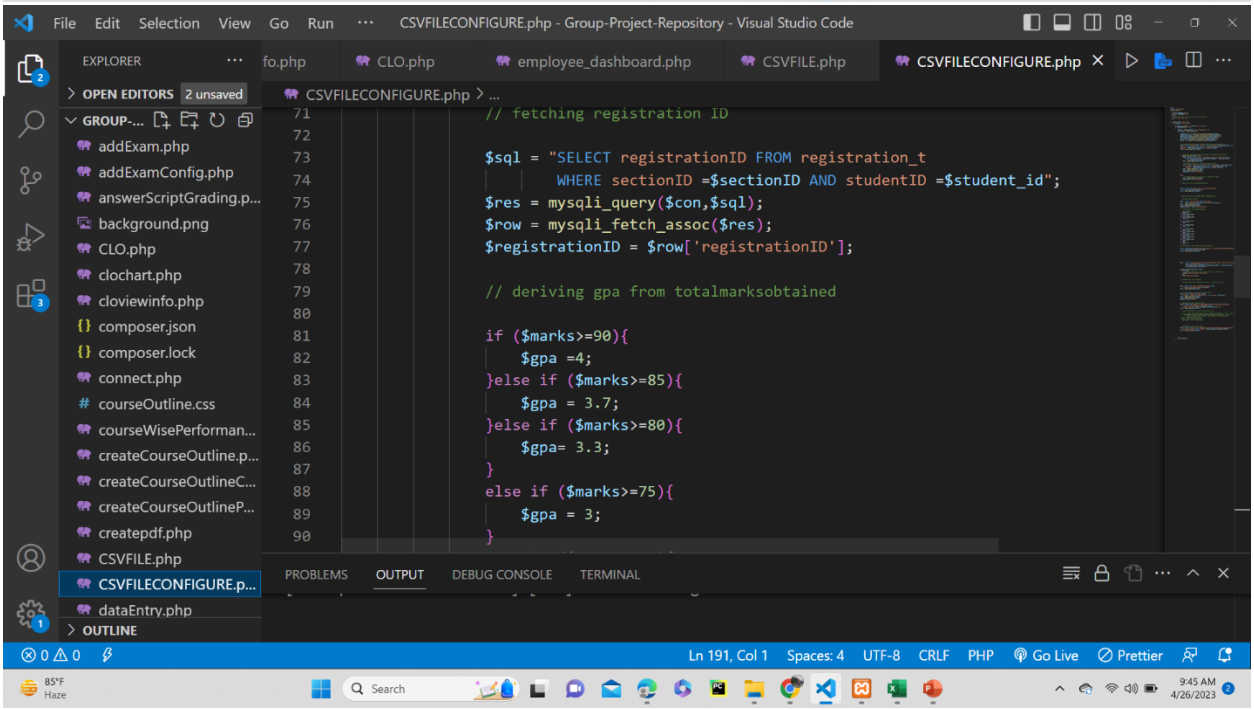

File Edit Selection View Go Run
CSVFILE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
studentcloinfo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILE.php > html > body > div.nav
<h1>SPMS 4.0</h1>
<form action="CSVFILECONFIGURE.php" method="POST" enctype='multipart/form-data'>
<label for="student-id">CSV FILE:</label>
<input type="file" id="file" name="file" accept=".csv" />
<input type="submit" value="Import" name ="import" id="submit">
</form>
<footer class="footer">
Student Performance Monitoring System
</footer>
</body>
</html>
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 83, Col 11 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

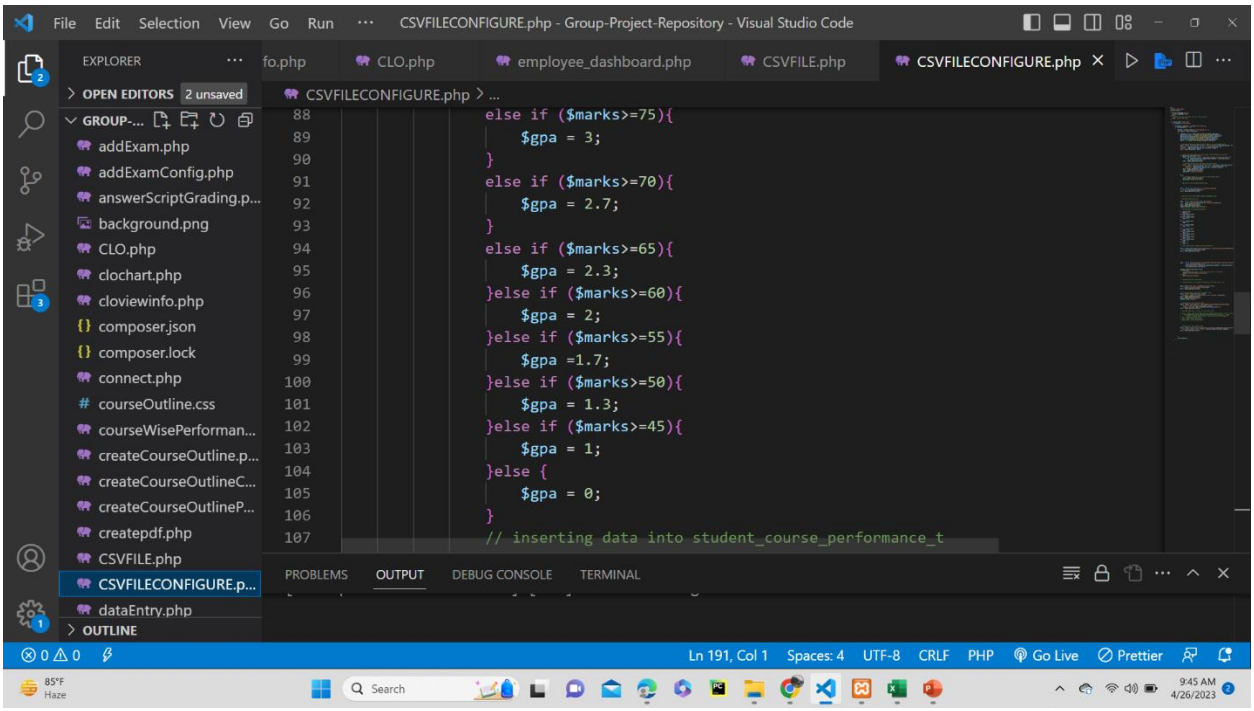

File Edit Selection View Go Run
CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
<?php
include 'connect.php';
session_start();
if(isset($_SESSION['ID'])){
$f_ID = $_SESSION['ID'];
}else{
// handle the case when $ID is not set in the session
echo 'f_ID is not set';
}
if (isset($_POST["import"])){
if ($_FILES['file']['name'])
{
$filename = explode(".",$_FILES['file']['name']);
if ($filename[1] == 'csv')
{
$handle = fopen($_FILES['file']['tmp_name'],"r");
while($data = fgetcsv($handle))
{
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

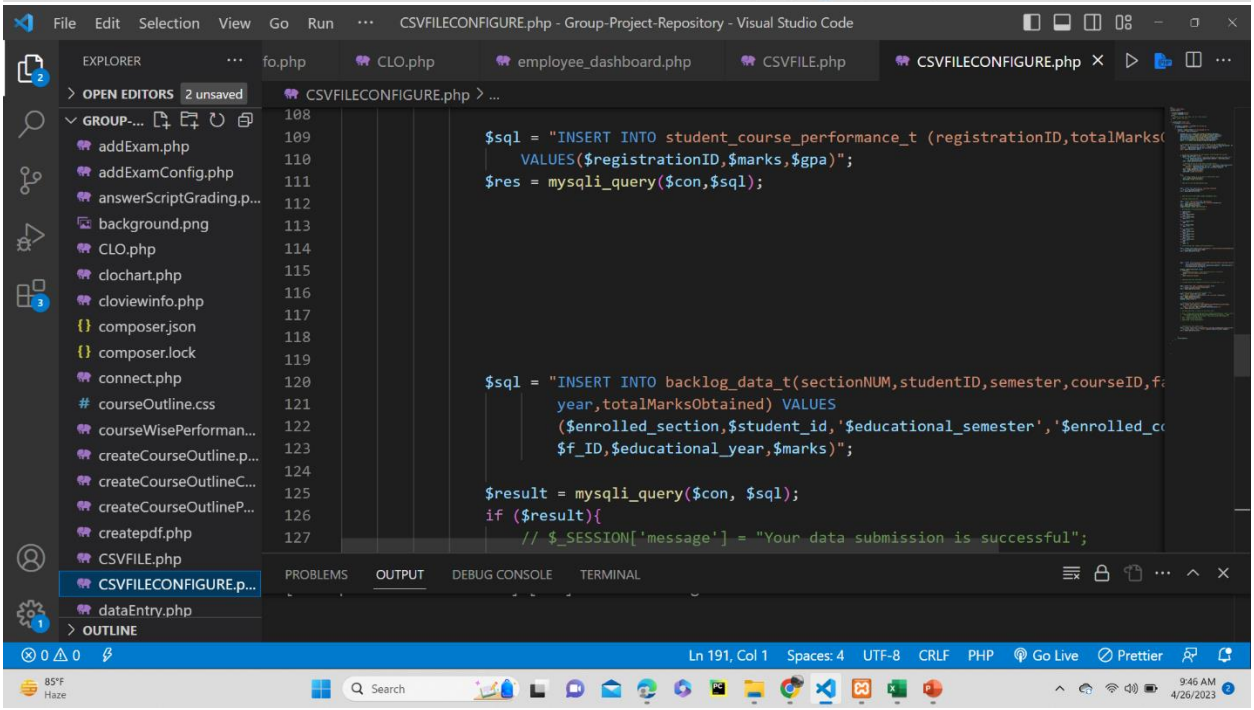

File Edit Selection View Go Run CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
while($data = fgetcsv($handle))
{
$student_id =(int) mysqli_real_escape_string($con,$data[0]);
$educational_year=(int)mysqli_real_escape_string($con,$data[1]);
$educational_semester=mysqli_real_escape_string($con,$data[2]);
$enrolled_course=mysqli_real_escape_string($con,$data[3]);
$enrolled_section=mysqli_real_escape_string($con,$data[4]);
$marks = (int)mysqli_real_escape_string($con,$data[5]);
// SEARCHING WHETHER SECTION IS ALREADY PRESENT IN THE DATABASE OR NOT
$sql = "SELECT sectionID FROM section_t where sectionNum ='$enrolled_sectio
courseID = '$enrolled_course' AND year = '$educational_year' AND
semester = '$educational_semester'AND facultyID ='$f_ID'";
$res = mysqli_query($con, $sql);
// IF SECTION IS NOT PRESENT IN THE DATABASE, WE ARE ADDING THAT SECTION
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

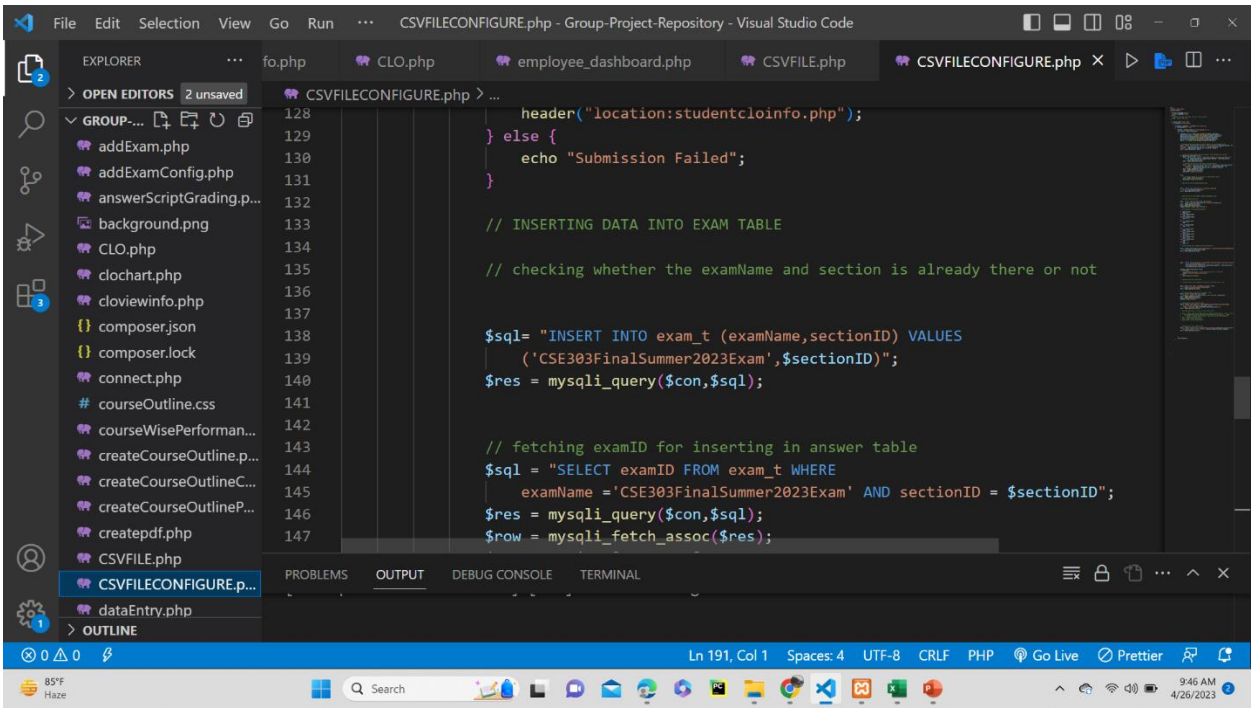

File Edit Selection View Go Run CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
// IF SECTION IS NOT PRESENT IN THE DATABASE, WE ARE ADDING THAT SECTION
if (mysqli_num_rows($res) == 0) {
$sql = "INSERT INTO section_t (sectionNum,semester,courseID,facultyID,y
VALUES ($enrolled_section,'$educational_semester','$enrolled_course
$f_ID,$educational_year)";
$res = mysqli_query($con,$sql);
// fetching section Id of the newly inserted section
$sql = "SELECT sectionID FROM section_t where sectionNum ='$enrolled_se
courseID = '$enrolled_course' AND year = '$educational_year' AND
semester = '$educational_semester'AND facultyID ='$f_ID'";
$res = mysqli_query($con, $sql);
$row = mysqli_fetch_assoc($res);
$sectionID = $row['sectionID'];
}
else{
// fetching section id to use fill up registration table
$row = mysqli_fetch_assoc($res);
$sectionID = $row['sectionID'];
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

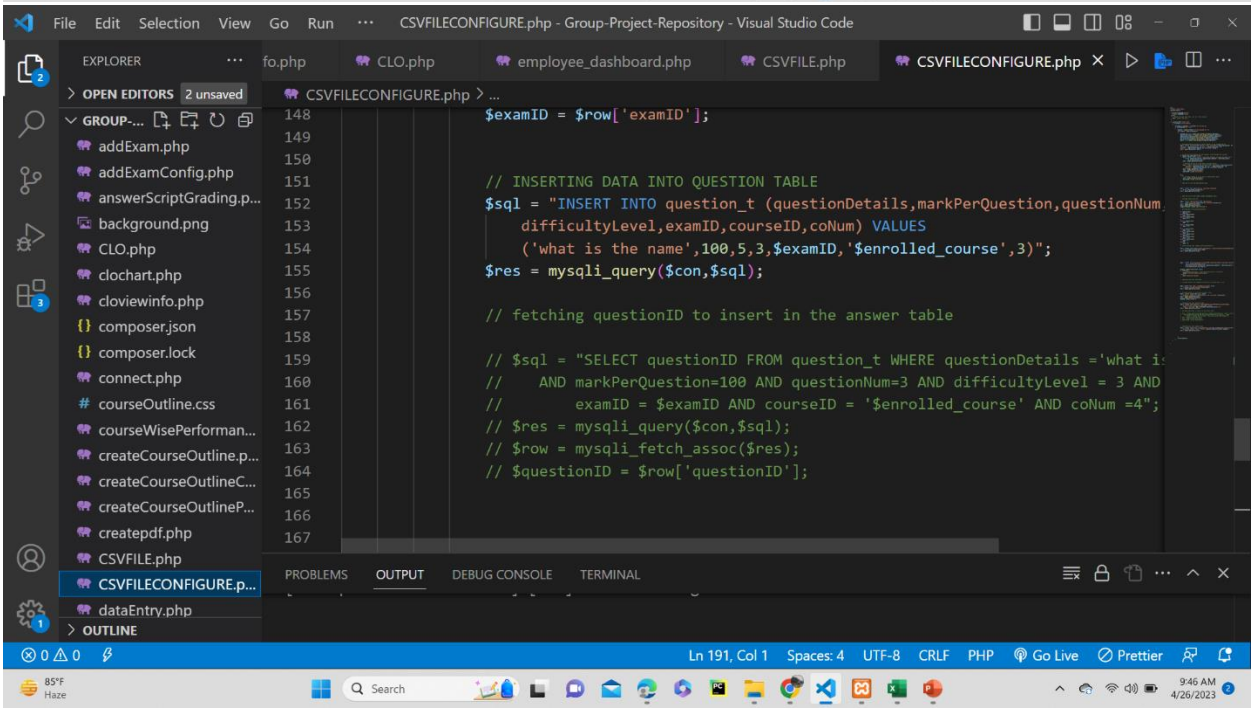

File Edit Selection View Go Run ··· CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
53 eise{
54 // fetching section id to use fill up registration table
55 $row = mysqli_fetch_assoc($res);
56 $sectionID = $row['sectionID'];
57 }
58
59 // CODE FOR FILL UP THE REGISTRATION TABLE
60
61
62
63 $sql = "INSERT INTO registration_t (sectionID,studentID)
64 VALUES ($sectionID,$student_id)";
65 $res = mysqli_query($con,$sql);
66
67
68
69 // CODE FOR FILL UP THE STUDENT COURSE PERFORMANCE TABLE
70
71 // fetching registration ID
72
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

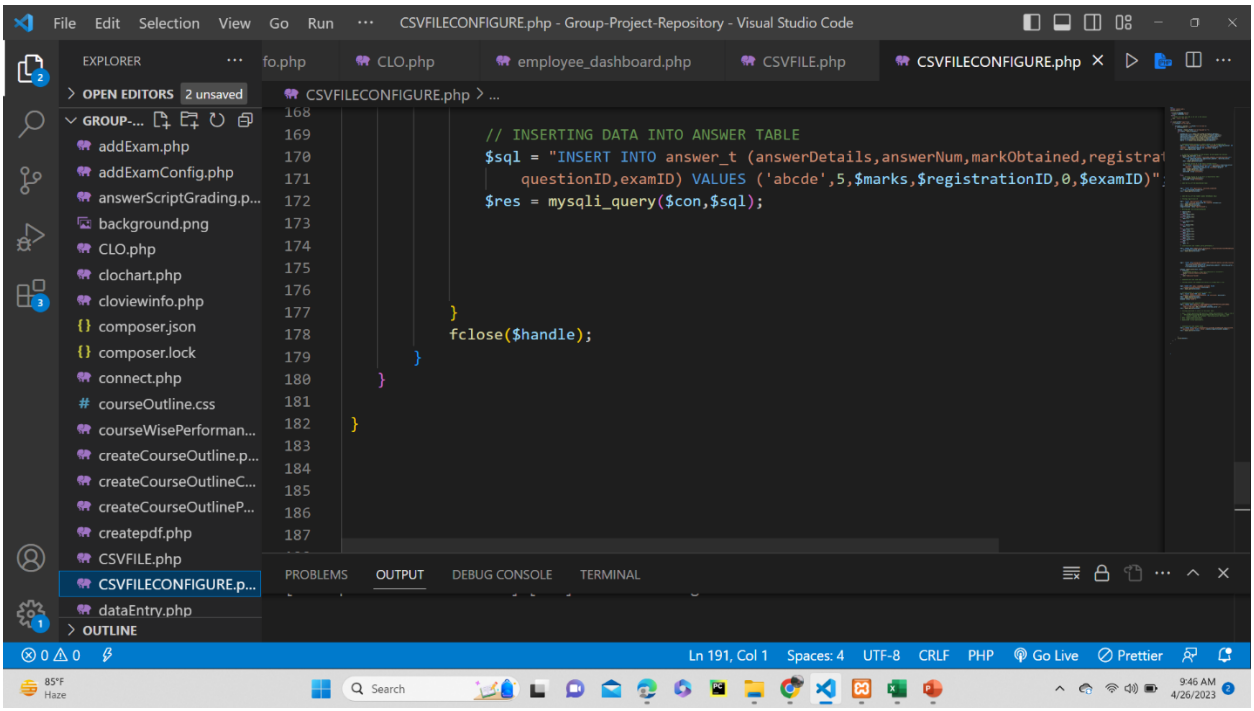

File Edit Selection View Go Run ··· CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
71 // fetching registration ID
72
73 $sql = "SELECT registrationID FROM registration_t
74 WHERE sectionID =$sectionID AND studentID =$student_id";
75 $res = mysqli_query($con,$sql);
76 $row = mysqli_fetch_assoc($res);
77 $registrationID = $row['registrationID'];
78
79 // deriving gpa from totalmarksobtained
80
81 if ($marks>=90){
82 $gpa =4;
83 }else if ($marks>=85){
84 $gpa = 3.7;
85 }else if ($marks>=80){
86 $gpa= 3.3;
87 }
88 else if ($marks>=75){
89 $gpa = 3;
90 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023

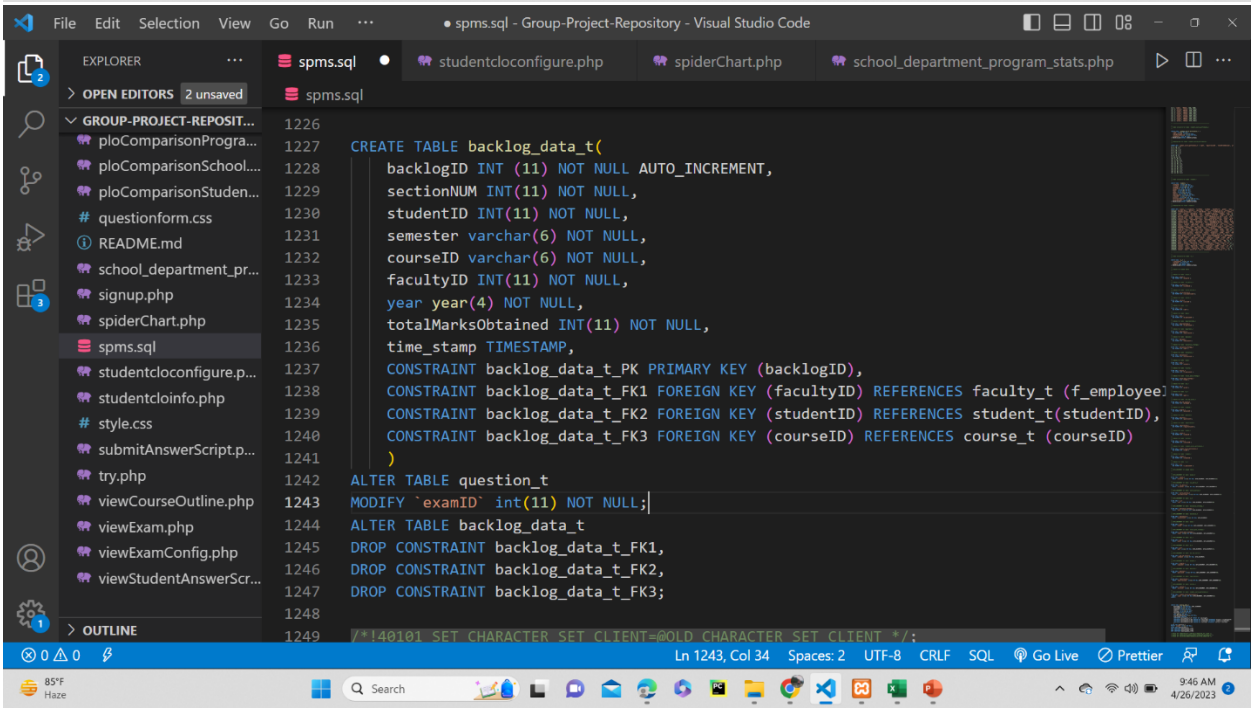

File Edit Selection View Go Run ··· CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
88 else if ($marks>=75){
89 $gpa = 3;
90 }
91 else if ($marks>=70){
92 $gpa = 2.7;
93 }
94 else if ($marks>=65){
95 $gpa = 2.3;
96 }else if ($marks>=60){
97 $gpa = 2;
98 }else if ($marks>=55){
99 $gpa =1.7;
100 }else if ($marks>=50){
101 $gpa = 1.3;
102 }else if ($marks>=45){
103 $gpa = 1;
104 }else {
105 $gpa = 0;
106 }
107 // inserting data into student_course_performance_t
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:45 AM 4/26/2023


File Edit Selection View Go Run ··· CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
108
109 $sql = "INSERT INTO student_course_performance_t (registrationID,totalMarks(
110 VALUES($registrationID,$marks,$gpa)";
111 $res = mysqli_query($con,$sql);
112
113
114
115
116
117
118
119
120 $sql = "INSERT INTO backlog_data_t(sectionNUM,studentID,semester,courseID,f
121 year,totalMarksObtained) VALUES
122 ($enrolled_section,$student_id,'$educational_semester','$enrolled_c
123 $f_ID,$educational_year,$marks)";
124
125 $result = mysqli_query($con, $sql);
126 if ($result){
127 // $_SESSION['message'] = "Your data submission is successful";
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:46 AM 4/26/2023


File Edit Selection View Go Run ··· CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
header("location:studentcloinfo.php");
} else {
echo "Submission Failed";
}
// INSERTING DATA INTO EXAM TABLE
// checking whether the examName and section is already there or not
$sql= "INSERT INTO exam_t (examName,sectionID) VALUES
('CSE303FinalSummer2023Exam',$sectionID)";
$res = mysqli_query($con,$sql);
// fetching examID for inserting in answer table
$sql = "SELECT examID FROM exam_t WHERE
examName ='CSE303FinalSummer2023Exam' AND sectionID = $sectionID";
$res = mysqli_query($con,$sql);
$row = mysqli_fetch_assoc($res);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:46 AM 4/26/2023


File Edit Selection View Go Run ··· CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
$examID = $row['examID'];
// INSERTING DATA INTO QUESTION TABLE
$sql = "INSERT INTO question_t (questionDetails,markPerQuestion,questionNum,
difficultyLevel,examID,courseID,coNum) VALUES
('what is the name',100,5,3,$examID,'$enrolled_course',3)";
$res = mysqli_query($con,$sql);
// fetching questionID to insert in the answer table
// $sql = "SELECT questionID FROM question_t WHERE questionDetails ='what i
// AND markPerQuestion=100 AND questionNum=3 AND difficultyLevel = 3 AND
// examID = $examID AND courseID = '$enrolled_course' AND coNum =4";
// $res = mysqli_query($con,$sql);
// $row = mysqli_fetch_assoc($res);
// $questionID = $row['questionID'];
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:46 AM 4/26/2023


File Edit Selection View Go Run CSVFILECONFIGURE.php - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-...
addExam.php
addExamConfig.php
answerScriptGrading.p...
background.png
CLO.php
clochart.php
cloviewinfo.php
composer.json
composer.lock
connect.php
courseOutline.css
courseWisePerforman...
createCourseOutline.p...
createCourseOutlineC...
createCourseOutlineP...
createpdf.php
CSVFILE.php
CSVFILECONFIGURE.p...
dataEntry.php
OUTLINE
fo.php
CLO.php
employee_dashboard.php
CSVFILE.php
CSVFILECONFIGURE.php
CSVFILECONFIGURE.php > ...
// INSERTING DATA INTO ANSWER TABLE
$sql = "INSERT INTO answer_t (answerDetails,answerNum,markObtained,registrat
questionID,examID) VALUES ('abcde',5,$marks,$registrationID,0,$examID)"
$res = mysqli_query($con,$sql);
}
fclose($handle);
}
}
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Ln 191, Col 1 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
85°F Haze
Search
9:46 AM 4/26/2023


File Edit Selection View Go Run • spms.sql - Group-Project-Repository - Visual Studio Code
EXPLORER
OPEN EDITORS 2 unsaved
GROUP-PROJECT-REPOSIT...
ploComparisonProgra...
ploComparisonSchool....
ploComparisonStuden...
questionform.css
README.md
school_department_pr...
signup.php
spiderChart.php
spms.sql
studentcloconfigure.p...
studentcloinfo.php
style.css
submitAnswerScript.p...
try.php
viewCourseOutline.php
viewExam.php
viewExamConfig.php
viewStudentAnswerScr...
OUTLINE
spms.sql
studentcloconfigure.php
spiderChart.php
school_department_program_stats.php
spms.sql
CREATE TABLE backlog_data_t(
backlogID INT (11) NOT NULL AUTO_INCREMENT,
sectionNUM INT(11) NOT NULL,
studentID INT(11) NOT NULL,
semester varchar(6) NOT NULL,
courseID varchar(6) NOT NULL,
facultyID INT(11) NOT NULL,
year year(4) NOT NULL,
totalMarksObtained INT(11) NOT NULL,
time_stamp TIMESTAMP,
CONSTRAINT backlog_data_t_PK PRIMARY KEY (backlogID),
CONSTRAINT backlog_data_t_FK1 FOREIGN KEY (facultyID) REFERENCES faculty_t (f_employee
CONSTRAINT backlog_data_t_FK2 FOREIGN KEY (studentID) REFERENCES student_t(studentID),
CONSTRAINT backlog_data_t_FK3 FOREIGN KEY (courseID) REFERENCES course_t (courseID)
)
ALTER TABLE question_t
MODIFY `examID` int(11) NOT NULL;
ALTER TABLE backlog_data_t
DROP CONSTRAINT backlog_data_t_FK1,
DROP CONSTRAINT backlog_data_t_FK2,
DROP CONSTRAINT backlog_data_t_FK3;
/*!40101 SET CHARACTER SET CLIENT=@OLD_CHARACTER_SET_CLIENT */;
Ln 1243, Col 34 Spaces: 2 UTF-8 CRLF SQL Go Live Prettier
85°F Haze
Search
9:46 AM 4/26/2023

CHAPTER 5 – CONCLUSION:

In conclusion, in the student performance monitoring system project we developed some valuable features that have improved the system's functionality and usability. With the inclusion of CLOs in the database, students can now track their progress in achieving specific learning objectives through an easy-to-understand chart. Additionally, the ability to create course outlines and question banks within the system has made it easier for teachers to manage their courses and provide relevant assessments.

Furthermore, students can now download PDFs directly from the system, enabling them to access important materials at any time, from anywhere. These enhancements have made the system more user-friendly, efficient, and effective in supporting students' learning and development.

Overall, this project has achieved its goal of enhancing the student performance monitoring system and providing a more comprehensive tool for tracking and improving student performance. The features we have implemented will undoubtedly have a positive impact on students' academic success and contribute to their future achievements.

A. PROBLEM AND SOLUTION:

Analysis Phase:

During the 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 in order to overcome such confusions, and information received during the interview was collected in order to get a better understanding of the system that was being developed.

Designing Phase:

Some problems were faced while creating the EERD and Relational Schema during the

Design Phase, However, constant feedbacks from the faculty were enough to overcome those issues.

Implementation Phase:
All the System Requirements were completed successfully.

Front-End Developing tools: HTML, CSS, JavaScript

Back-End Developing tools: PHP.

Database-integration: MySQL

B. Additional Features and Future Development:

In SPMS 4.0, we have already implemented features like tracing the student po and co. We have also introduced some features like creating course outline, questions bank, and evaluating exam script. Students can see their course wise performance, course wise co mapping. Faculty and admin also can see these features in their end.

There are still a lot of features we can include in our system to make our system more useful to students. as well as faculty. Some features are mentioned below, we can add in our system in future:

- In the system we can include a feature that enables parents and teachers to communicate with each other about student progress. This could be particularly useful for students who may need additional support outside of the classroom.
- The system can include a feature where both students and faculty can see the students. attendance.
- In the system we also can add a system where faculty would be able to trace the students who are going to fail the course by analyzing their mid and quiz exam. This system will be very helpful for the students as the teacher would be able to help those students by seeing their performance. We also can add a feature like faculty will get a notification that in his/her course, who are performing poorly.
- The system could provide multilingual support to ensure that students who speak languages other than the primary language of instruction are able to fully engage with the system.