

Course Management System:

A software program called a course management system is created to assist with administration, teachers, and students. It offers a platform for interaction and information sharing about the course and module content, grades, and other topics between instructors and students.

This software is built using the help of Java Swing Framework in Eclipse and using XAMPP for the backend database. A total of 10 java files and 1 database file consisting of 4 tables were implemented for the completion of the task.

Access Modifiers:

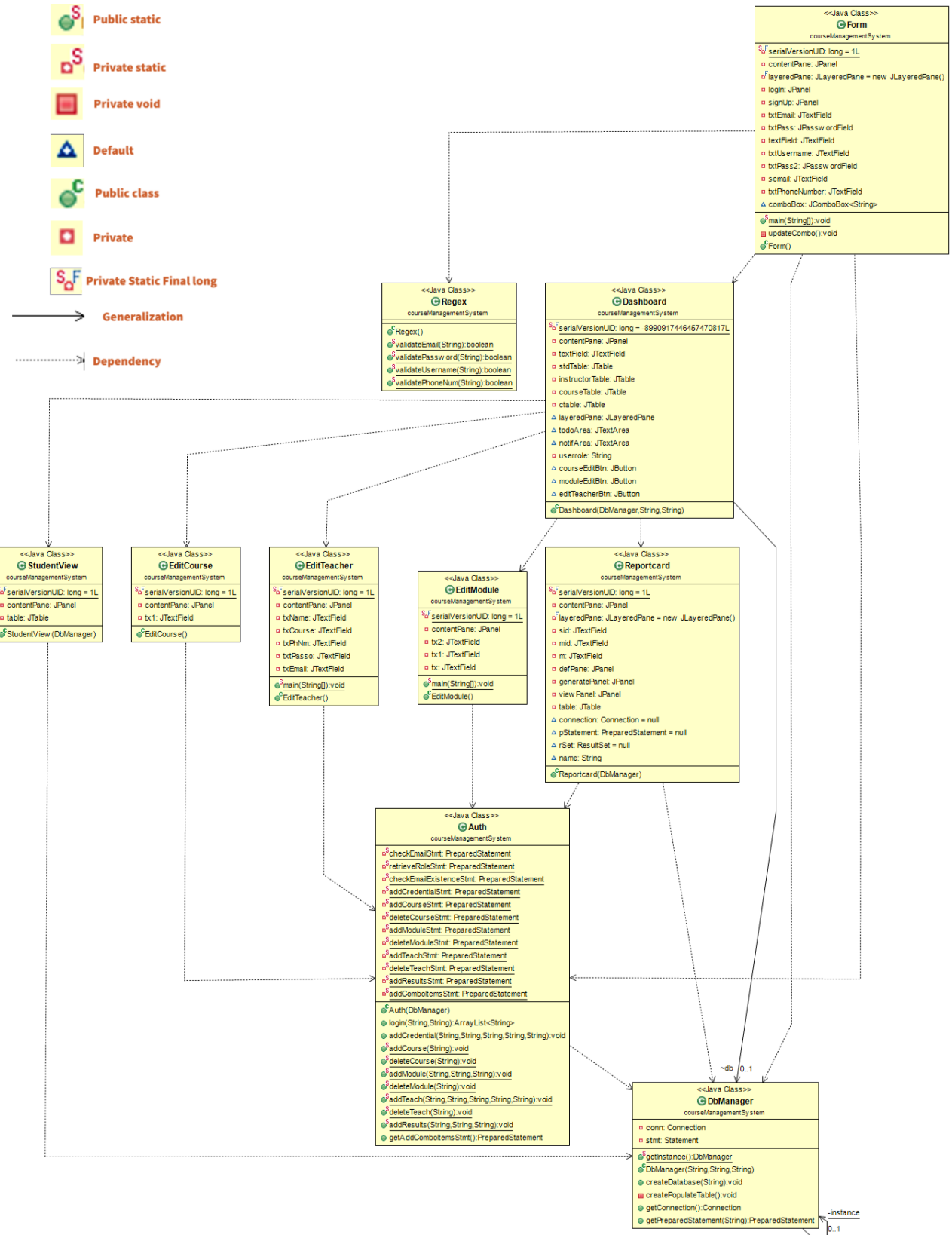
private: Accessible only within the class.

default: Accessible only within the package.

protected: Accessible within the package and outside the package but through inheritance only.

public: Accessible everywhere

UML Class Diagram:



1. Form.java:

It involves the login and sign-up for the course management system. Access modifiers are 12 private members and one default member. It has a public main method along with one private void function. It has a main class Form.

It depends on Dashboard, Regex, Auth and DbManager.java.

2. Dashboard.java:

It is the main body of the application. Mostly all activities are carried out from here. Access modifiers are 8 private members along with 6 default members. It has a main class Dashboard() which takes the instance, name and role.

It depends on StudentView, EditCourse, EditTeacher, EditModule and Reportcard.java. It has an "is-a" relationship with DbManager.java.

3. DbManager.java:

It is the back body of the application. It carries out the creation and insertion of databases and tables. Access modifiers are 2 private members. It has a main class DbManager() which takes the path, username and password of the XAMPP server. It has one public static method along with one private void method and 3 public void methods.

4. Auth.java:

It is the body which contains all the statements and methods that are carried out in the application. Access modifiers are 12 private static members having one main class Auth() which takes db from the DbManager.java. It has 7 public static methods and 3 public methods.

It depends on DbManager.java.

5. Reportcard.java:

It is where the teacher can view and generate the report. Access modifiers are 8 private members along with one final and 4 default members. It contains a main class Reportcard() which takes db from the DbManager.java.

It depends on Auth, DbManager.java.

6. StudentView.java:

It is where the student can view the report. Access modifiers are 2 private members and have one main class StudentView() which takes db from the DbManager.java.

It depends on DbManager.java.

7. Regex.java:

It is the security guard for unwanted entries. It has a main class Regex() and contains 3 public static methods.

8. EditCourse:

Here the user who has admin privileges can edit the course. Access modifiers are 3 private members along with one main class EditCourse().

It depends on Auth.java.

9. EditModule:

Here the user who has admin privileges can edit the module. Access modifiers are 5 private members along with one main method and main class EditModule().

It depends on Auth.java.

10. EditTeacher:

Here the user who has admin privileges can edit the teacher. Access modifiers are 7 private members along with one main class EditTeacher().

It depends on Auth.java.

Overall, the use of course management system application can serve to simplify and enhance the management of courses, facilitating communication and information sharing between instructors and students as well as monitoring and evaluating course performance for administrators.

Screenshots of XAMPP:

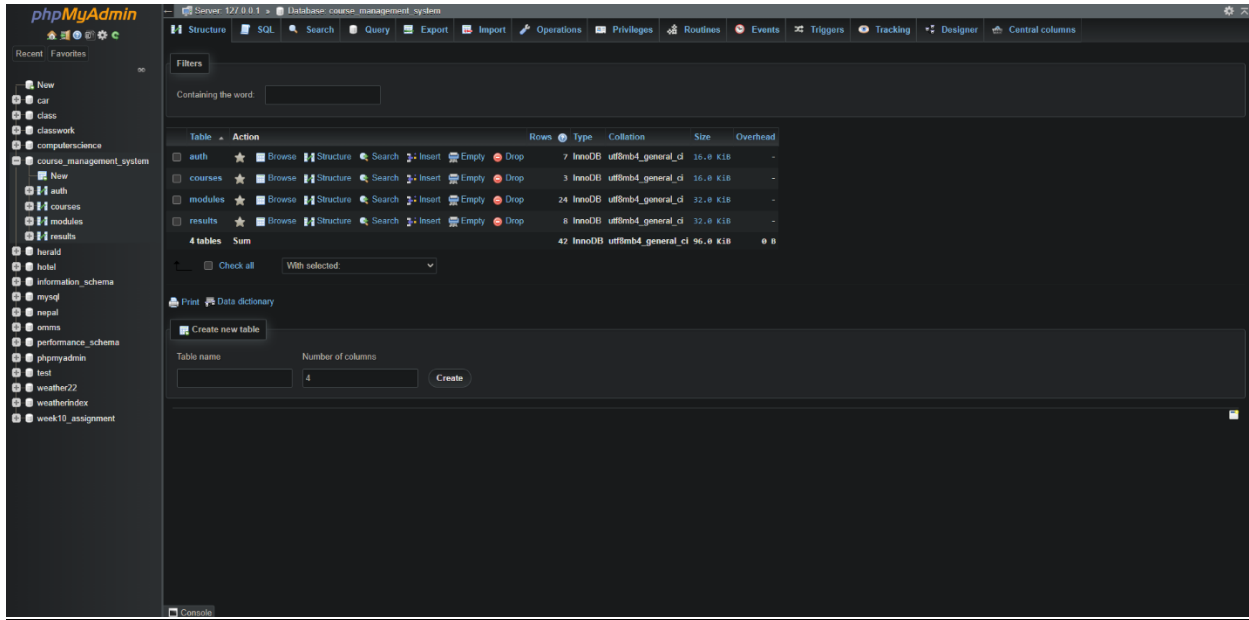


Figure 1. Database

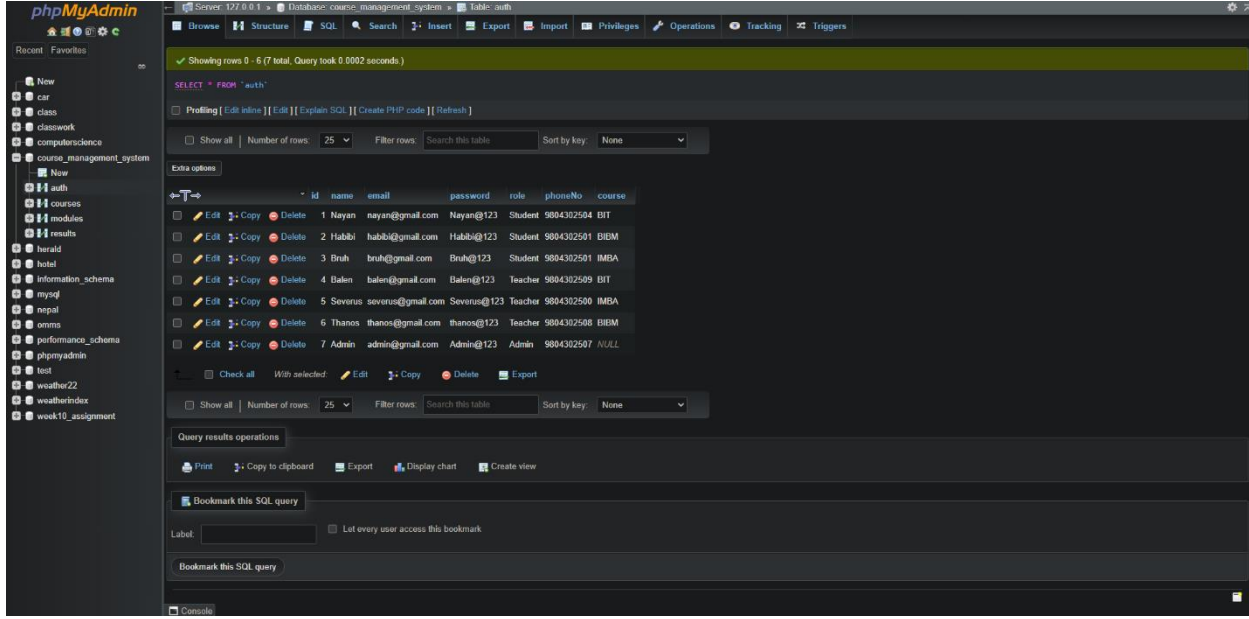


Figure 2. auth table

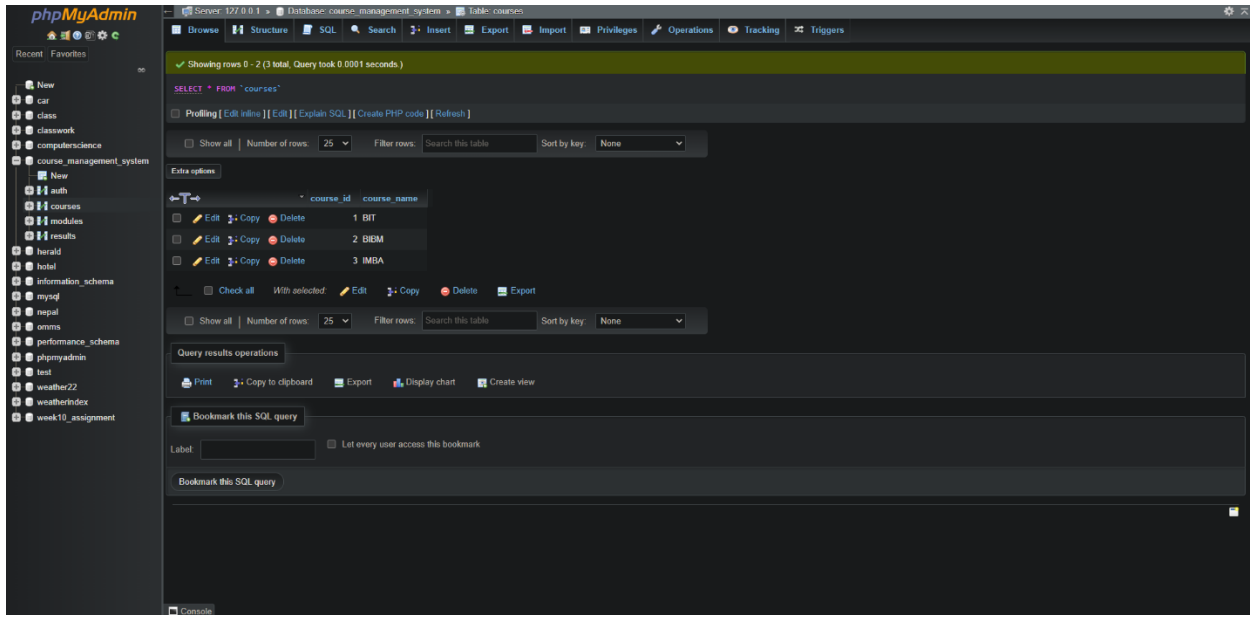


Figure 3. courses table

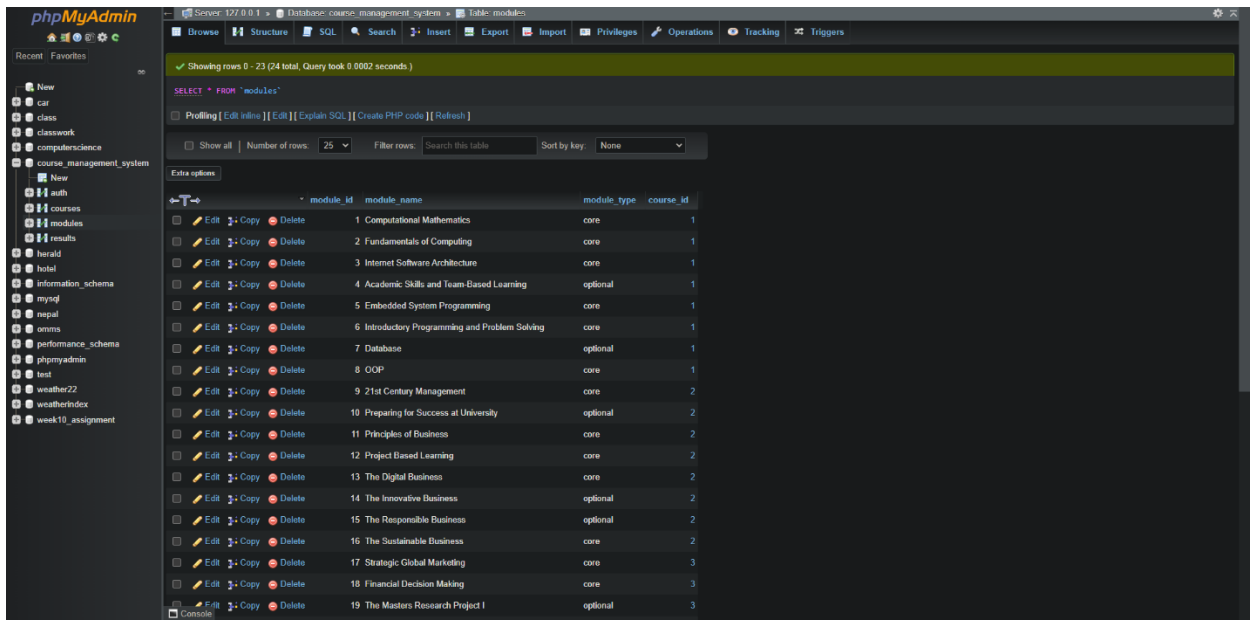


Figure 4. modules table

The screenshot shows the phpMyAdmin interface with a table named 'results' displayed. The table has three columns: 'student_id', 'module_id', and 'marks'. The data is as follows:

student_id	module_id	marks
1	1	99
1	2	1
1	3	40
1	4	75
1	5	88
1	6	39
1	7	41
1	8	100

The interface includes a sidebar with a tree view of databases and tables, a top navigation bar with options like 'Browse', 'Structure', and 'SQL', and a main content area with a query editor and table controls. The query shown is 'SELECT * FROM `results`'. Below the table, there are options for 'Query results operations' such as 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. There is also a 'Bookmark this SQL query' section with a label input field and a checkbox for 'Let every user access this bookmark'.

Figure 5. results table