CP476 Project

Group 2:
Jessie Newman, Phoebe Schulman,
Arzekeil-Abel Garcia De Leon, Yasmine Ali

Introduction - Purpose

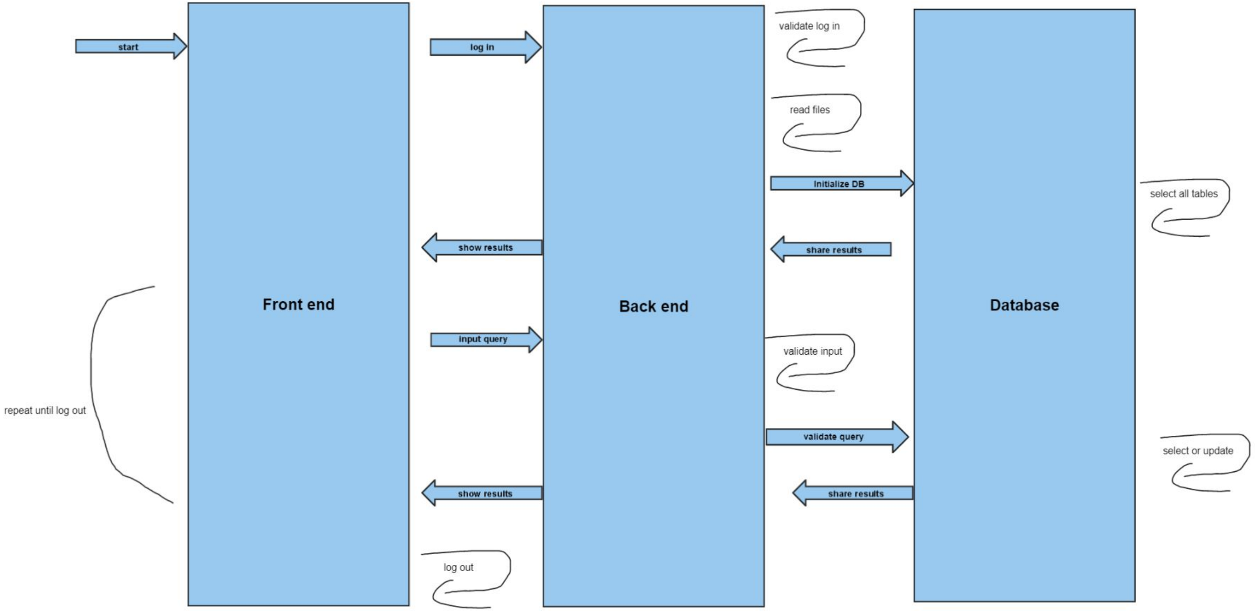
- Main purpose:
    - Develop a web server
    - Users can interact with a database server
- 2 large goals:
    - Create and display database tables
        - Input: Name Table and Course Table
        - Output: Student Final Grade Table
    - **Execute SQL statements given by web users**
        - SELECT
        - UPDATE

Introduction - Purpose

- Uses SQL, HTML, and PHP
- Apache: web server
- MySQL: database server

Introduction - System Overview

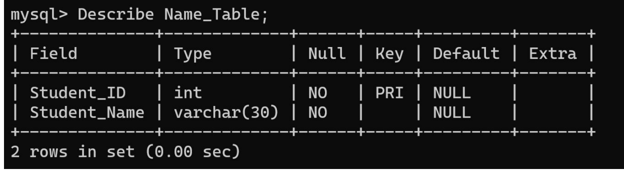
- 3 components
  - Database, frontend and backend



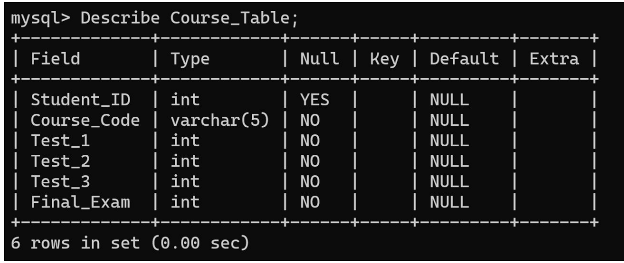
Database: Idea

- Create a database with **3 tables**
  - Name Table and Course Table
    - Created from 2 files
  - Student Final Grade Table
    - Based on 2 input tables

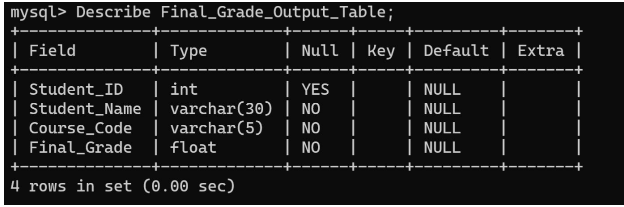
- Schemas
  - Primary key: Student_ID



```
mysql> Describe Name_Table;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Student_ID   | int         | NO   | PRI | NULL    |       |
| Student_Name | varchar(30) | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



```
mysql> Describe Course_Table;
+-------------+------------+------+-----+---------+-------+
| Field       | Type       | Null | Key | Default | Extra |
+-------------+------------+------+-----+---------+-------+
| Student_ID  | int        | YES  |     | NULL    |       |
| Course_Code | varchar(5) | NO   |     | NULL    |       |
| Test_1      | int        | NO   |     | NULL    |       |
| Test_2      | int        | NO   |     | NULL    |       |
| Test_3      | int        | NO   |     | NULL    |       |
| Final_Exam  | int        | NO   |     | NULL    |       |
+-------------+------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

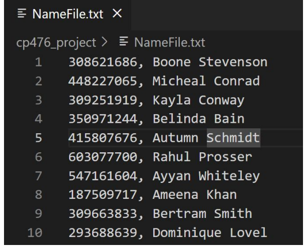


```
mysql> Describe Final_Grade_Output_Table;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Student_ID   | int         | YES  |     | NULL    |       |
| Student_Name | varchar(30) | NO   |     | NULL    |       |
| Course_Code  | varchar(5)  | NO   |     | NULL    |       |
| Final_Grade  | float       | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

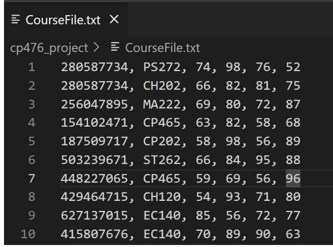
Database: Approach

- Connect to MySQL database
- Create 1 empty database
  - To later store 3 tables
  - $sql = "**CREATE DATABASE** IF NOT EXISTS " . DATABASE_NAME;
- Connect to new CP476_Database
- Create schemas for 3 empty tables
  - Name Table, Course Table, and Student Final Grade Table
  - $sql= "**CREATE TABLE** ". NAME_TABLE_NAME . "(Student_ID INT(9) PRIMARY KEY, Student_Name VARCHAR(30) NOT NULL)";

Database: Approach



```
308621686, Boone Stevenson
448227065, Micheal Conrad
309251919, Kayla Conway
350971244, Belinda Bain
415807676, Autumn Schmidt
603077700, Rahul Prosser
547161604, Ayyan Whiteley
187509717, Ameena Khan
309663833, Bertram Smith
293688639, Dominique Lovel
```



```
280587734, PS272, 74, 98, 76, 52
280587734, CH202, 66, 82, 81, 75
256047895, MA222, 69, 80, 72, 87
154102471, CP465, 63, 82, 58, 68
187509717, CP202, 58, 98, 56, 89
503239671, ST262, 66, 84, 95, 88
448227065, CP465, 59, 69, 56, 96
429464715, CH120, 54, 93, 71, 80
627137015, EC140, 85, 56, 72, 77
415807676, EC140, 70, 89, 90, 63
```

- Insert data for input tables
    - Read files: NameFile.txt and CourseFile.txt
        - Traverse files line by line
        - Words are separated by commas
    - Insert into table

```
$sql = $connection->prepare("INSERT INTO ". NAME_TABLE_NAME . "(Student_ID, Student_Name) VALUES(?,?)");
$sql ->bind_param("ss", $nameStudentID, $studentName);
$sql->execute();
```

Database: Approach

- Insert data for output table (Student Final Grade Table)
    - **1: select everything from COURSE_TABLE and NAME_TABLE**
        - **Joined on Student_ID**

```
$sql = "SELECT C.Student_ID, N.Student_Name, C.Course_Code, C.Test_1,
C.Test_2, C.Test_3, C.Final_Exam
FROM " . COURSE_TABLE_NAME . " AS C
INNER JOIN ". NAME_TABLE_NAME . " AS N
ON C.Student_ID = N.Student_ID";
```

Database: Approach

- Insert data for output table (Student Final Grade Table)
    - 2: calculate formula using the input tables:

      finalGrade = test1 * 0.20 + test2 * 0.20 + test3 * 0.20 + finalExam * 0.40
    - 3: insert data of each row
        - **INSERT INTO**

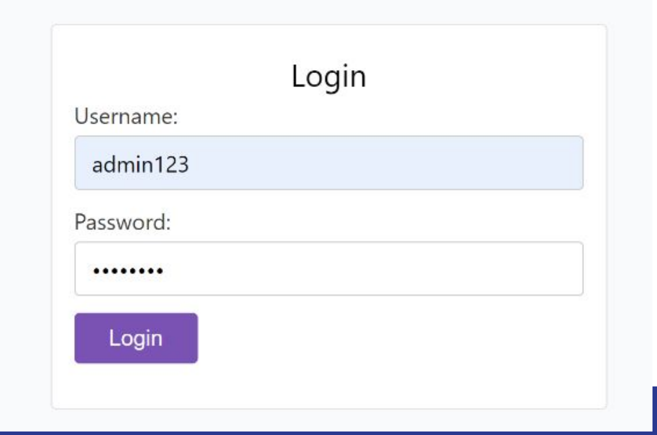
- Close the connection to our database

Front End: Idea

- Allows user interaction
    - Login page
        - To securely access the database
    - **Input SQL queries**
        - Textbox for SELECT/UPDATE
        - Submit button
    - **Display resulting tables**
    - Logout button

Program Approach: (Front End) log in /out

- HTML form used to receive POST REQUEST
    - <form action = “cp476_project.php” method = post>
- SESSION variables to store authentication information across multiple pages
- Begin/resume session on every page accessed through authentication
- Destroy session and redirect to login page upon logging out



Login

Username:

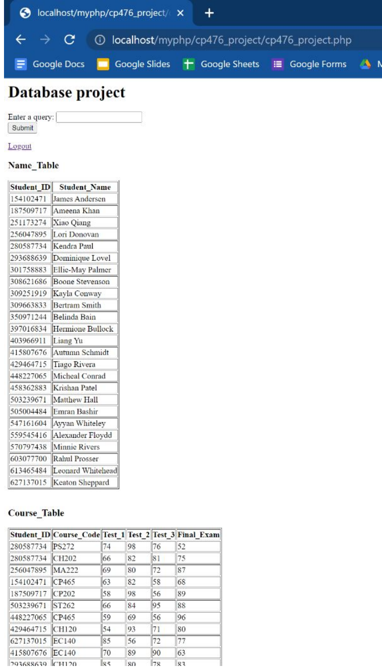
admin123

Password:

•••••••••

Program Approach: (Front End) HTML forms

- Page 1: display all tables and wait for first input
- SELECT * FROM each table
- Input box
    - **<input type ="text"** name ="query">
- Submit button: go to page 2
    - **<input type = "submit"** name = "show databases">
- Logout button
    - Links to logout.php, which returns to login page


localhost/myphp/cp476_project/
localhost/myphp/cp476_project/cp476_project.php
Google Docs
Google Slides
Google Sheets
Google Forms
Submit

Database project

Enter a query:

Logout

Name_Table

Student_ID	Student_Name
154102471	James Andersen
187509717	Ameena Khan
251173274	Xiao Qiang
256047895	Lori Donovan
280587734	Kendra Paul
293688639	Dominique Lovel
301758883	Ellie-May Palmer
308621686	Boone Stevenson
309251919	Kayla Conway
309663833	Bertram Smith
350971244	Belinda Bain
397016834	Hermione Bullock
403966911	Liang Yu
415807676	Autumn Schmidt
429464715	Tiago Rivera
448227065	Micheal Conrad
458362883	Krishan Patel
503239671	Matthew Hall
505004484	Emran Bashir
547161604	Ayyan Whiteley
559545416	Alexander Floydd
570797438	Minnie Rivers
603077700	Rahul Prosser
613465484	Leonard Whitehead
627137015	Keaton Sheppard

Course_Table

Student_ID	Course_Code	Test_1	Test_2	Test_3	Final_Exam
280587734	PS272	74	98	76	52
280587734	CH202	66	82	81	75
256047895	MA222	69	80	72	87
154102471	CP465	63	82	58	68
187509717	CP202	58	98	56	89
503239671	ST262	66	84	95	88
448227065	CP465	59	69	56	96
429464715	CH120	54	93	71	80
627137015	EC140	85	56	72	77
415807676	EC140	70	89	90	63
293688639	CH120	85	80	78	83

Program Approach: (Front End) HTML forms

- Page 2: display results and wait for next input
- Display results
    - Similar to display all tables
- Wait for next input
    - Similar to wait for first input

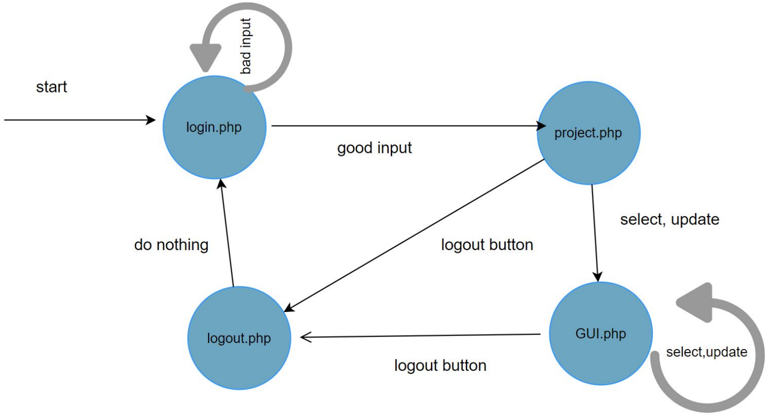
Back End: Idea

- **Validates user input**
  - Query starts with select or update
  - Valid SQL statement

Program Approach: (Back End) input processing

- Access user input, submitted from webpage
  - Front end:
    - <form action = "cp476_project_GUI.php" method = "**post**">
    - <input type ="text" name ="**query**">
  - Back end:
    - $userInput = **$_POST['query'];**

File Structure



- Cp476_project_login
  - Frontend: login page
- Cp476_project
  - Backend: initialize database and tables
    - NameFile.txt and CourseFile.txt as input
  - Frontend: display all tables and wait for first input
- Cp476_project_GUI
  - Backend: processes user input
  - Frontend: display results and wait for next input
- Cp476_project_logout
  - Frontend: return to login page
-

Demonstration: local host and command prompt

- Command prompt

```
mysql -u root -p

drop database cp476_database;

Show databases;
```

Demonstration: local host and command prompt

- Log into local host: http://localhost/myphp/cp476_project/cp476_project_login.php

    - Query the database: select students in MA222

select Student_ID, Course_Code from Course_Table where Course_Code = "MA222";

Demonstration: local host and command prompt

- Command prompt
- Show the database (after log in)

Show databases;

Use cp476_database;

- Query the database: first select
  - Same result

select Student_ID, Course_Code from Course_Table where Course_Code = "MA222";

Demonstration: local host and command prompt

- Local host
  - Query the database: update

```
UPDATE course_table SET student_id  = 88 WHERE Student_ID = 256047895;
```

  - Query the database: select (again)

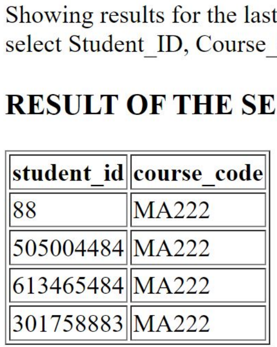
```
select Student_ID, Course_Code from Course_Table where Course_Code = "MA222";
```

Demonstration: local host and command prompt

- Command prompt
- Select again

select Student_ID, Course_Code from Course_Table where Course_Code = "MA222";

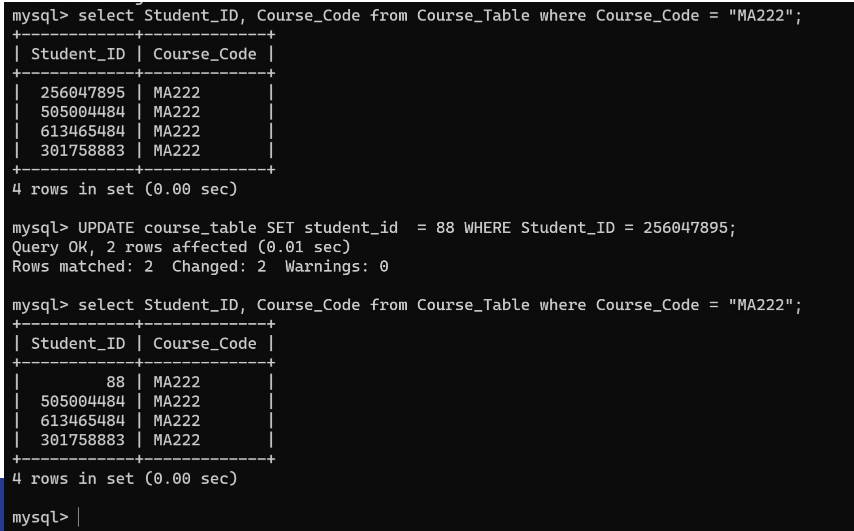
- **Same query results**



Showing results for the last query:
select Student_ID, Course_Code from Course_Table where Course_Code = "MA222";

RESULT OF THE SELECT IS

student_id	course_code
88	MA222
505004484	MA222
613465484	MA222
301758883	MA222



```
mysql> select Student_ID, Course_Code from Course_Table where Course_Code = "MA222";
+------------+-------------+
| Student_ID | Course_Code |
+------------+-------------+
|  256047895 | MA222       |
|  505004484 | MA222       |
|  613465484 | MA222       |
|  301758883 | MA222       |
+------------+-------------+
4 rows in set (0.00 sec)

mysql> UPDATE course_table SET student_id  = 88 WHERE Student_ID = 256047895;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select Student_ID, Course_Code from Course_Table where Course_Code = "MA222";
+------------+-------------+
| Student_ID | Course_Code |
+------------+-------------+
|         88 | MA222       |
|  505004484 | MA222       |
|  613465484 | MA222       |
|  301758883 | MA222       |
+------------+-------------+
4 rows in set (0.00 sec)

mysql> 
```

Demonstration: local host and command prompt

- Local host
  - Error: invalid SQL statements

select t from Course_Table;

  - Error: valid SQL statements, other than select or update

INSERT INTO Name_Table (Student_id, Student_Name) VALUES (4006, 'peter');

- Log out

Sources

Kaplarevic, V. (2021, August 12). *How to Drop a Table in MySQL*. Knowledge Base by phoenixNAP. https://phoenixnap.com/kb/mysql-drop-table

MySQL DROP DATABASE - How to Delete a Database in MySQL Server. (2021, July 26). MySQL Tutorial. https://www.mysqltutorial.org/mysql-drop-database/

L. (2019, October 10). *List (Show) Tables in a MySQL Database*. Linuxize. https://linuxize.com/post/show-tables-in-mysql-database/

PHP: explode - Manual. (n.d.). https://www.php.net/manual/en/function.explode.php

PHP MySQL Select Data. (n.d.). https://www.w3schools.com/php/php_mysql_select.asp

Show values from a MySQL database table inside a HTML table on a webpage. (n.d.). Stack Overflow. https://stackoverflow.com/questions/17902483/show-values-from-a-mysql-database-table-inside-a-html-table-on-a-webpage

Sources

CrowdStrike. (2022, November 9). *What is a SQL Injection Attack? | CrowdStrike*. crowdstrike.com. https://www.crowdstrike.com/cybersecurity-101/sql-injection/#:~:text=SQL%20injection%20(SQLi)%20is%20a,application%20security%20risk%20in%202021

Hanna, K. T., & Lewis, S. (2021, June 28). *SQL injection*. Software Quality. https://www.techtarget.com/searchsoftwarequality/definition/SQL-injection#:~:text=A%20SQL%20injection%20is%20a,or%20execute%20malicious%20SQL%20statements

Online Causal Loop Diagram Tool. (n.d.). https://online.visual-paradigm.com/diagrams/features/causal-loop-diagram-tool/

PHP mysqli fetch_all() Function. (n.d.). https://www.w3schools.com/php/func_mysqli_fetch_all.asp

PHP mysqli fetch_array() Function. (n.d.). https://www.w3schools.com/php/func_mysqli_fetch_array.asp

PHP mysqli query() Function. (n.d.). https://www.w3schools.com/php/func_mysqli_query.asp

Sources

K. (n.d.). *How to get input field value using PHP*. Edureka Community. https://www.edureka.co/community/94514/how-to-get-input-field-value-using-php#:~:text=Use%20PHP's%20%24_POST%20or%20%24_GET,name%20of%20the%20HTML%20tag.&text=To%20show%20the%20value%3A,%24_GET%5B'subject'%5D%3B%20%3F%3E

PHP mysqli fetch_fields() Function. (n.d.). https://www.w3schools.com/php/func_mysqli_fetch_fields.asp

PHP MySQLi Functions. (n.d.). https://www.w3schools.com/php/php_ref_mysqli.asp

PHP: strtolower - Manual. (n.d.). https://www.php.net/manual/en/function.strtolower.php

SQL UPDATE Statement. (n.d.). https://www.w3schools.com/sql/sql_update.asp

What do strict types do in PHP? (n.d.). Stack Overflow. https://stackoverflow.com/questions/48723637/what-do-strict-types-do-in-php

PHP Tutorial. (2021, September 29). *PHP Login*. https://www.phptutorial.net/php-tutorial/php-login/

SQL INSERT INTO Statement. (n.d.). https://www.w3schools.com/sql/sql_insert.asp

Q and A