Week 4 – Database Development & Class Registration

Anthony Meza

University of Arizona Global Campus

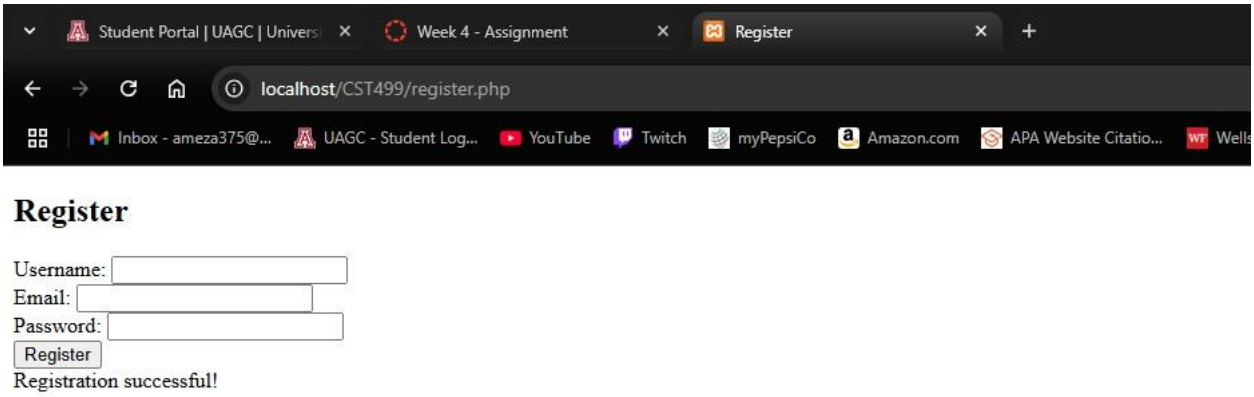
CST 499: Capstone for Computer Software Technology

Instructor Charmelia Butler

November 4, 2024

Database Development & Class Registration

This week, I continued to re-develop a dynamic web application to manage class registrations, connecting a MySQL database to a user interface. This implementation phase involved setting up backend infrastructure, creating functional pages, and establishing CRUD operations for a seamless registration system.



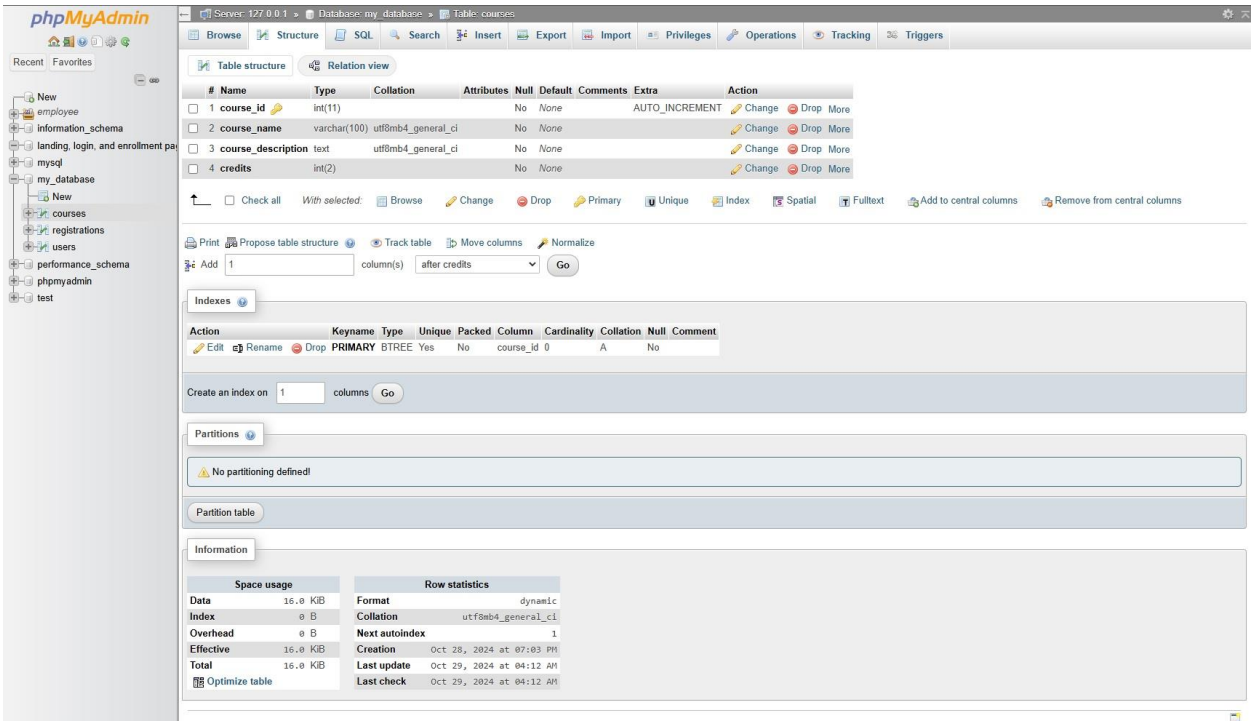
Summary of the Implementation Phase

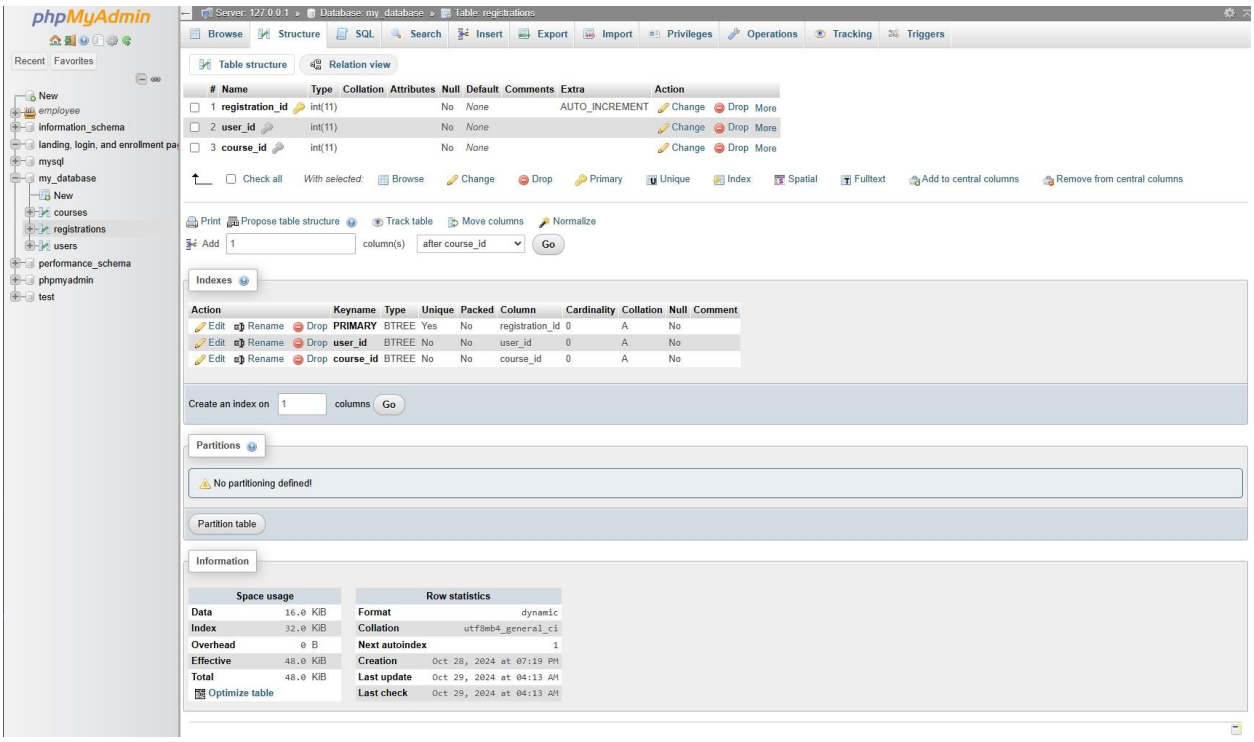
The primary goal of this phase was to create the foundation for user registration and class management functions. The project required setting up multiple components, including database configuration, page connections, and essential features like class registration, listing, and deletion.

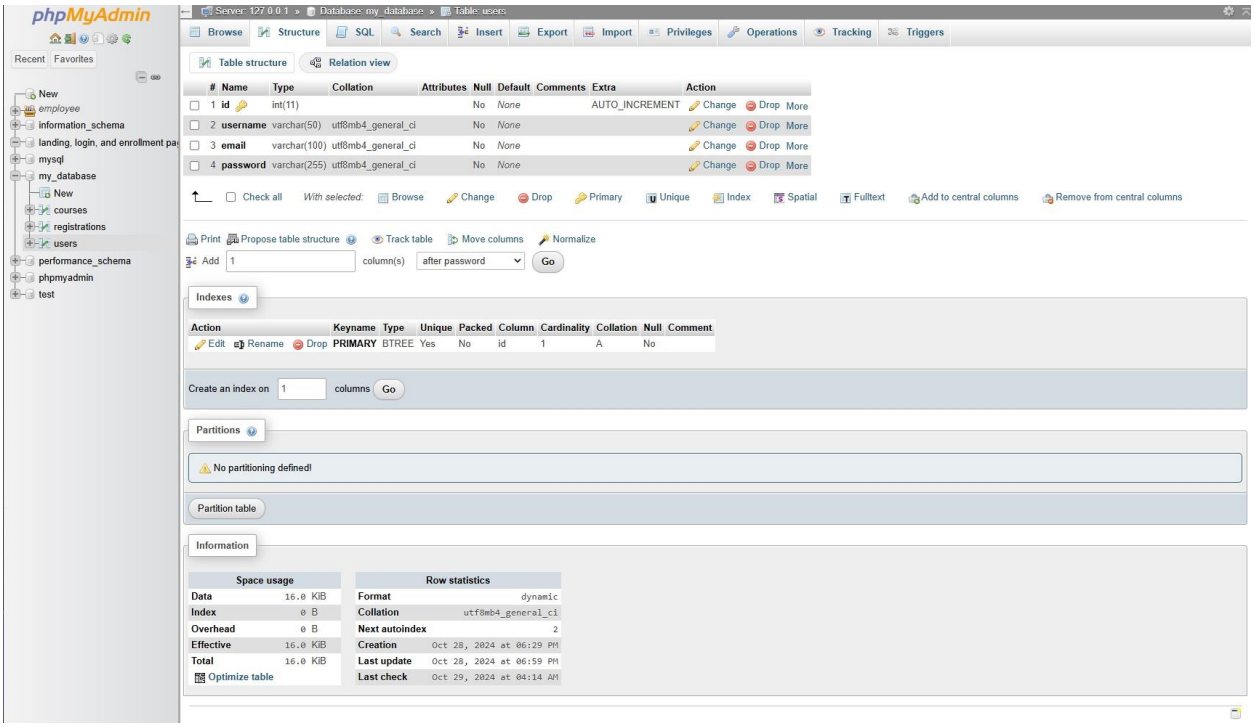
Experience with Database Development

Creating the MySQL database for this project was both challenging and rewarding. I started by defining the necessary tables: users, courses, and registrations. Each table's attributes, relationships,

and constraints were carefully considered to ensure data consistency and integrity. I used foreign keys to link the users table with registrations and courses, supporting a many-to-many relationship between students and classes. Using AUTO_INCREMENT and PRIMARY KEY constraints helped me ensure unique identifiers for each entry, streamlining data management and retrieval.


phpMyAdmin
Recent Favorites
New
employee
information_schema
landing, login, and enrollment pa
mysql
my_database
New
courses
registrations
users
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: my_database » Table: courses
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 course_id int(11) No None AUTO_INCREMENT Change Drop More
2 course_name varchar(100) utf8mb4_general_ci No None Change Drop More
3 course_description text utf8mb4_general_ci No None Change Drop More
4 credits int(2) No None Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to central columns Remove from central columns
Print Propose table structure Track table Move columns Normalize
Add 1 column(s) after credits Go
Indexes
Action Keyname Type Unique Packed Column Cardinality Collation Null Comment
Edit Rename Drop PRIMARY BTREE Yes No course_id 0 A No
Create an index on 1 columns Go
Partitions
No partitioning defined!
Partition table
Information
Space usage
Data 16.0 KiB
Index 0 B
Overhead 0 B
Effective 16.0 KiB
Total 16.0 KiB
Optimize table
Row statistics
Format dynamic
Collation utf8mb4_general_ci
Next autoindex 1
Creation Oct 28, 2024 at 07:03 PM
Last update Oct 29, 2024 at 04:12 AM
Last check Oct 29, 2024 at 04:12 AM

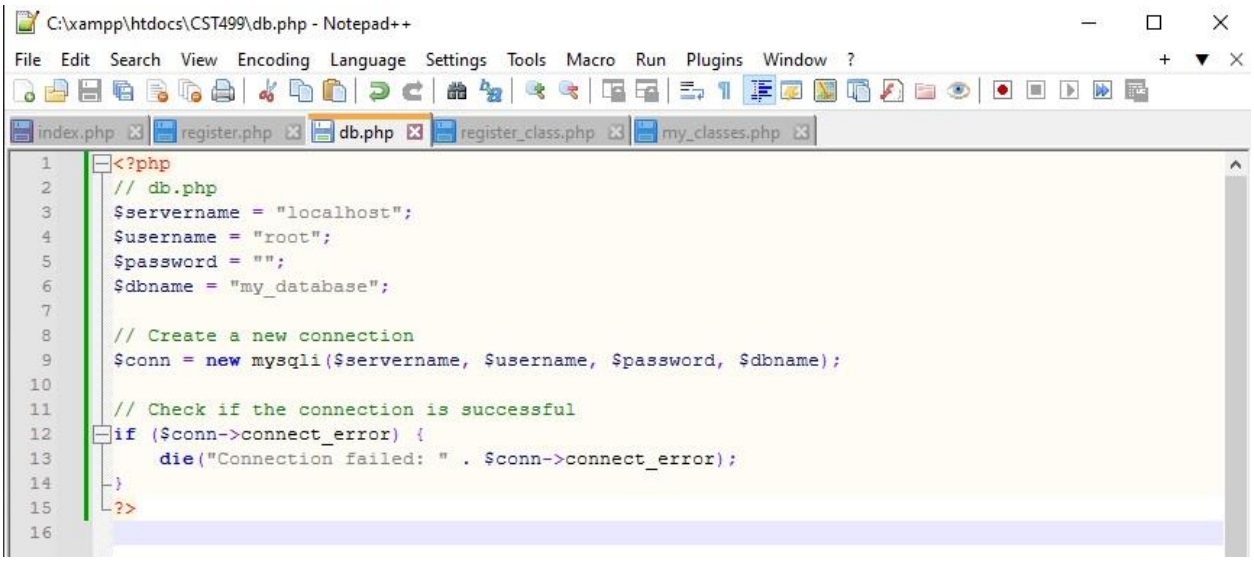

phpMyAdmin
Recent Favorites
New
employee
information_schema
landing, login, and enrollment pa
mysql
my_database
New
courses
registrations
users
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: my_database » Table: registrations
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Table structure Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 registration_id int(11) No None AUTO_INCREMENT Change Drop More
2 user_id int(11) No None Change Drop More
3 course_id int(11) No None Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext Add to central columns Remove from central columns
Print Propose table structure Track table Move columns Normalize
Add 1 column(s) after course_id Go
Indexes
Action Keyname Type Unique Packed Column Cardinality Collation Null Comment
Edit Rename Drop PRIMARY BTREE Yes No registration_id 0 A No
Edit Rename Drop user_id BTREE No No user_id 0 A No
Edit Rename Drop course_id BTREE No No course_id 0 A No
Create an index on 1 columns Go
Partitions
No partitioning defined!
Partition table
Information
Space usage
Data 16.0 KiB
Index 32.0 KiB
Overhead 0 B
Effective 48.0 KiB
Total 48.0 KiB
Optimize table
Row statistics
Format dynamic
Collation utf8mb4_general_ci
Next autoindex 1
Creation Oct 28, 2024 at 07:19 PM
Last update Oct 29, 2024 at 04:13 AM
Last check Oct 29, 2024 at 04:13 AM



One learning point during this step was understanding foreign key constraints and how they enable referential integrity, ensuring each registered class links correctly to a user and course.

Connecting Pages to the Backend

Establishing a centralized database connection using a db.php file simplified the workflow by letting me connect all PHP pages to the MySQL database without repetitive code. Including this file on each page helped manage the connection across the application, reducing potential errors. I gained insight into how a well-structured backend connection file enhances scalability and reduces repetitive code in larger projects.



Developing Class Registration, Listing, and Deletion Functions

The registration page was particularly insightful as I worked on implementing a user-friendly form that displayed courses as checkboxes, allowing students to register for multiple classes simultaneously. When a user selected classes and submitted the form, PHP code handled inserting each registration entry into the registrations table, a process that emphasized the importance of looping logic in data entry.

C:\xampp\htdocs\CST499\my_classes.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.php | register.php | db.php | register_class.php | my_classes.php

```php
<!-- my_classes.php -->
<?php
include 'db.php';
$user_id = 1; // Replace with the logged-in user's ID
$result = $conn->query("
    SELECT courses.course_name, courses.credits
    FROM registrations
    JOIN courses ON registrations.course_id = courses.course_id
    WHERE registrations.user_id = $user_id
");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Classes</title>
</head>
<body>
    <h2>Your Registered Classes</h2>
    <ul>
        <?php while ($class = $result->fetch_assoc()): ?>
            <li><?php echo $class['course_name']; ?> (<?php echo $class['credits']; ?> credits)<
        <?php endwhile; ?>
    </ul>
</body>
</html>
```

Listing classes with a delete option was also critical, as it demonstrated CRUD (Create, Read, Update, Delete) functionalities. The delete function required special attention, as I needed to create a form that included a hidden input to capture and pass the course_id for deletion. Testing and refining this functionality provided me with hands-on experience in building a user-friendly interface while maintaining backend integrity.

C:\xampp\htdocs\CST499\my_classes.php - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

index.php | register.php | db.php | register_class.php | my_classes.php

```php
<?php
include 'db.php'; // Connect to the database
$user_id = 1; // Replace with the logged-in user's ID

// Check if a course deletion is requested
if (isset($_POST['delete_course_id'])) {
    $course_id = $_POST['delete_course_id'];
    $conn->query("DELETE FROM registrations WHERE user_id = $user_id AND course_id = $course_id"
    echo "Course deleted successfully!";
}

// Query to fetch registered courses for the user
$result = $conn->query("
    SELECT courses.course_id, courses.course_name, courses.credits
    FROM registrations
    JOIN courses ON registrations.course_id = courses.course_id
    WHERE registrations.user_id = $user_id
");
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Classes</title>
</head>
<body>
    <h2>Your Registered Classes</h2>
    <ul>
        <?php while ($class = $result->fetch_assoc(): ?>
            <li>
                <?php echo $class['course_name']; ?> (<?php echo $class['credits']; ?> credits)
                <form method="post" style="display:inline;">
                    <input type="hidden" name="delete_course_id" value="<?php echo $class['cours
                    <button type="submit">Delete</button>
                </form>
            </li>
        <?php endwhile; ?>
    </ul>
</body>
</html>
```

Challenges and Resolutions

One of the main challenges was managing database connections consistently across pages. Resolving these issues involved debugging connection settings and testing each page independently. Another challenge was ensuring that data manipulation, particularly for deletion, did not accidentally remove the wrong data. Careful use of hidden fields and SQL conditions helped mitigate this risk.

Conclusion

This implementation phase provided practical experience in integrating databases with dynamic web pages. By working through each functionality from creating tables to setting up the class registration system I gained valuable insight into database-driven web development. Going forward, I look forward to expanding this project with additional features and further enhancing my skills in PHP and MySQL.