Montclair State University

Dragons & Databases

Jared Kruegel
kruegelj1@montclair.edu
Dhruv Rana
ranad17@montclair.edu
Xavier Warner
warnerx3@montclair.edu
Stavros Kokkinos
kokkinoss1@montclair.edu

CSIT-335-01
17 December 2023

Usernames and Passwords

This is the official database account that interacts with the system itself

- Username: warnerx3_binary
- Password: bionicle731

This account is a test 1 for a student account

- Username: xavier
- Password: xavier

This account is a test 2 for a student account

- Username: jared
- Password: jared

This account is a test 3 for a student account

- Username: dhruv
- Password: dhruv

This account is a test 4 for a student account

- Username: stavros
- Password: stavros

This is an official student account

- Username: John
- Password: john

This is an official admin account

- Username: test
- Password: password

Summary of Problem

Over the past few weeks, our team was tasked with creating a database using the skills we learned in class. The team, Database and Dragons, has four members and our goal was to use the game "Dungeons and Dragons" as inspiration to develop a course registration site and respective database. We were hired by the local tabletop game store "The Dragons Dungeon". For our project, we used Dungeons and Dragons 5th edition. We chose this edition because it was the newest addition and we realized that many participants in the store's annual DND parties were newcomers to the 5th edition. They only had a limited time to get hands-on experience while playing at the store. Our team's solution involves creating an online platform offering courses covering various aspects of D&D fifth edition. We provided these ideas to our client, and they agreed that our proposed idea was a good way to proceed.

Analysis of Client's Needs

Our courses cover fundamental topics, class guides, subclasses, and Dungeon Master classes. The target audience is individuals who are interested in learning about the world and the more technical side of DND's fifth edition. Our team envisioned a user-friendly platform with easy course sign-ups, a simple understanding of DND, and global accessibility. The implementation plan involves two user groups: admin and student both with specific privileges.

Admin Privileges:

- Create Admin Account
- Create a Course
- Confirm Course Completion
- Confirm of Credits

Student Privileges:

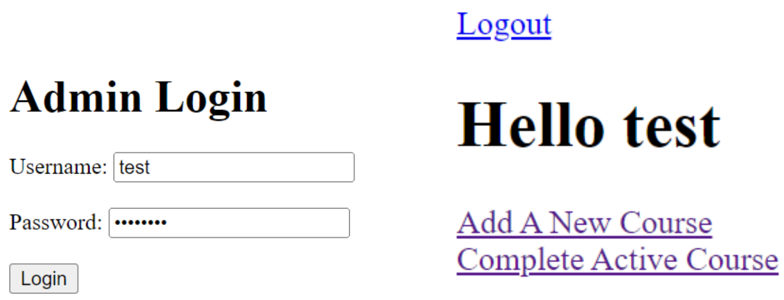
- Create Student Account
- See Credits Earned
- See Courses Completed
- See Profile Page
- Register for Courses

Our client requested that we implement 20 courses. These are the courses that we decided would be best to implement into the system.

- Learning the Basics
- Abilities and Skills
- Combat Basics
- Actions in Combat
- Barbarian Class Guide
- Bard Class Guide
- Cleric Class Guide
- Druid Class Guide
- Fighter Class Guide
- Monk Class Guide
- Paladin Class Guide

- Ranger Class Guide
- Rogue Class Guide
- Sorcerer Class Guide
- Warlock Class Guide
- Wizard Class Guide
- Artificer Class Guide
- Blood Hunter Class Guide
- Intro to Subclasses
- Dungeon Master Class

Discussion of Data Views & SQL Statements



Admin Login

Username: test

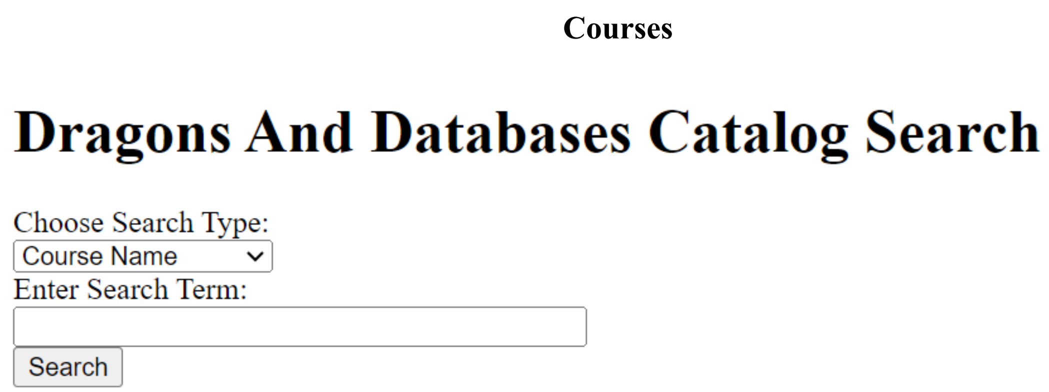
Password: ••••••••

Hello test

Figure 1: Admin Login

This is the SQL code to create the Administrator user. The SQL code creates the ‘admin_user’ table in a relational database with the attributes ‘id’, ‘userID’ which is the primary key, ‘admin_user_name’, and ‘admin_password’. The attributes ‘admin_user_name’ and ‘admin_password’ hold the Administrator’s username and password. The attribute ‘id’ is used to identify the administrator, and the ‘userID’ is used to identify the users on the system.

```
CREATE TABLE ‘admin_user’
(
  id bigint(13)
  userID bigint(13) not null primary key,
  admin_user_name varchar(100),
  admin_password varchar(100)
);

INSERT INTO ‘admin_user’ (‘id’, ‘userID’, ‘admin_user_name’, ‘admin_password’) VALUES
 (1, 1, "test", "password");
```



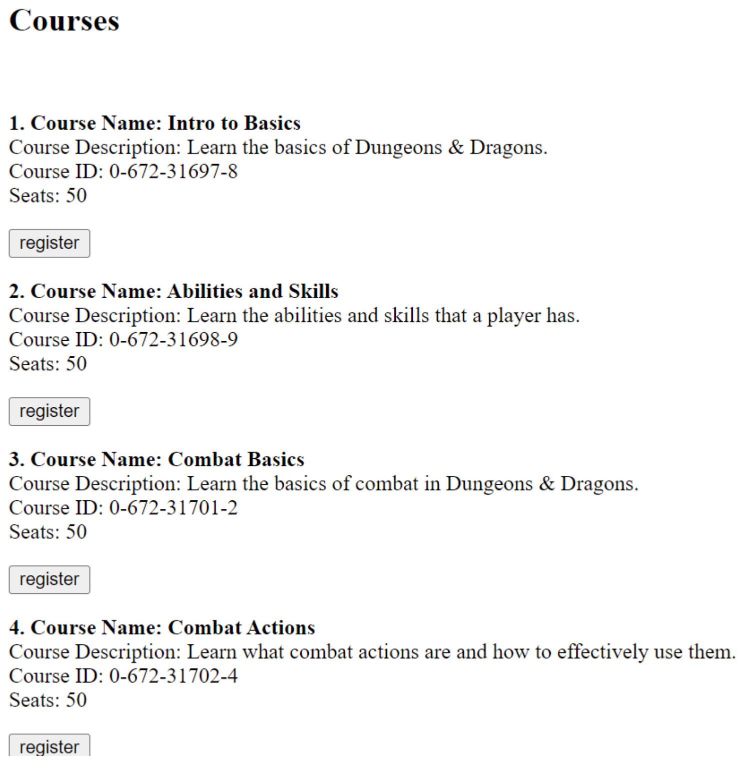
Courses

Dragons And Databases Catalog Search

Choose Search Type: Course Name

Enter Search Term:

Figure 2: Catalog Search



Courses

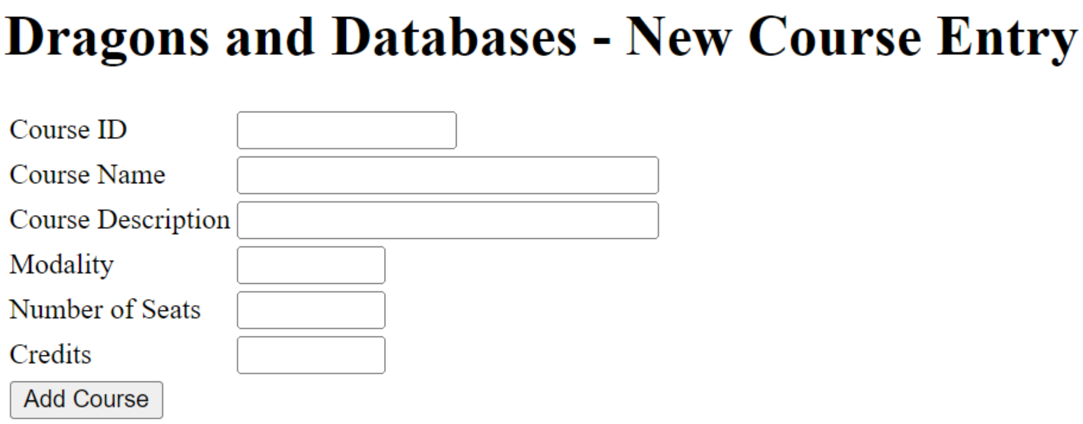
1. Course Name: Intro to Basics
Course Description: Learn the basics of Dungeons & Dragons.
Course ID: 0-672-31697-8
Seats: 50

2. Course Name: Abilities and Skills
Course Description: Learn the abilities and skills that a player has.
Course ID: 0-672-31698-9
Seats: 50

3. Course Name: Combat Basics
Course Description: Learn the basics of combat in Dungeons & Dragons.
Course ID: 0-672-31701-2
Seats: 50

4. Course Name: Combat Actions
Course Description: Learn what combat actions are and how to effectively use them.
Course ID: 0-672-31702-4
Seats: 50

Figure 3: Register for a Course



Dragons and Databases - New Course Entry

Course ID

Course Name

Course Description

Modality

Number of Seats

Credits

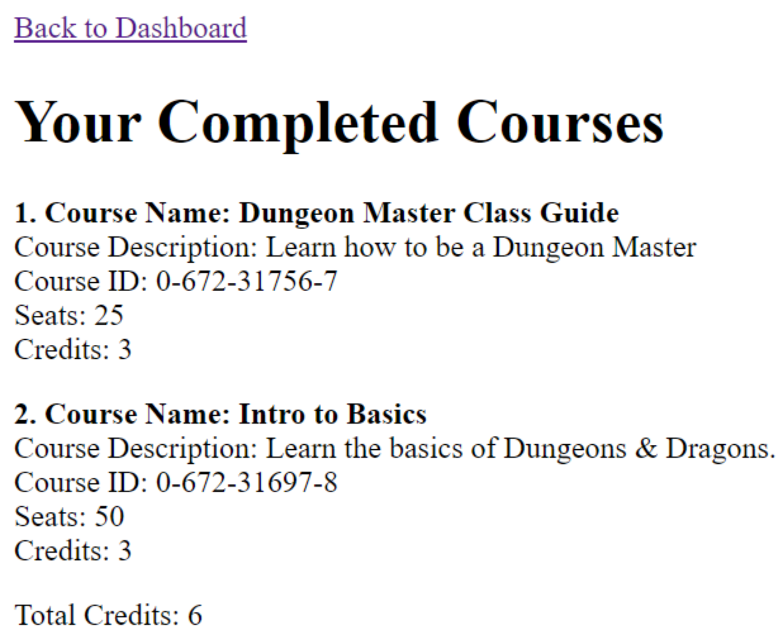
Figure 4: Enter a New Course

This is the SQL code to create the courses. The SQL code creates the ‘courses’ table in a relational database with the attributes ‘courseID’ which is the primary key, ‘topic’, ‘description’, ‘modality’, ‘numberStudents’, and ‘credits’. This table stores the details for the courses offered by Dragons and Database. The attributes ‘topic’ and ‘courseID’, along with ‘description’ are used as key options for searching for courses, showcasing the completion of a course, and adding a course.

```
CREATE TABLE ‘courses’
(  ‘courseID’ char(13) not null primary key,
   ‘topic’ char(50),
   ‘description’ char(100),
   ‘modality’ char(100),
   ‘numberStudents’ int(11),
   ‘credits’ int(11)
);
```

```
INSERT INTO ‘courses’ (‘courseID’, ‘topic’, ‘description’, ‘modality’, ‘numberStudents’, ‘credits’ VALUES
  ("0-672-31697-8", "Intro to Basics", "Learn the basics of Dungeons & Dragons.", "Online", 50, 3),
  ("0-672-31698-9", "Abilities and Skills", "Learn the abilities and skills that a player has.", "Online", 50, 3),
  ("0-672-31701-2", "Combat Basics", "Learn the basics of combat in Dungeons & Dragons.", "Online", 50, 3),
  ("0-672-31702-4", "Combat Actions", "Learn what combat actions are and how to effectively use them.", "Online", 50, 3),
  ("0-672-31703-5", "Barbarian Class Guide", "Learn how to play the Barbarian Class", "Online", 35, 3),
  ("0-672-31707-8", "Bard Class Guide", "Learn how to play the Bard Class", "Online", 35, 3),
  ("0-672-31711-1", "Cleric Class Guide", "Learn how to play the Cleric Class", "Online", 35, 3),
  ("0-672-31712-3", "Druid Class Guide", "Learn how to play the Druid Class", "Online", 35, 3),
  ("0-672-31714-6", "Fighter Class Guide", "Learn how to play the Fighter Class", "Online", 35, 3),
  ("0-672-31718-8", "Monk Class Guide", "Learn how to play the Monk Class", "Online", 35, 3),
  ("0-672-31724-2", "Paladin Class Guide", "Learn how to play the Paladin Class", "Online", 35, 3),
  ("0-672-31725-3", "Ranger Class Guide", "Learn how to play the Ranger Class", "Online", 35, 3),
  ("0-672-31728-4", "Rogue Class Guide", "Learn how to play the Rogue Class", "Online", 35, 3),
  ("0-672-31736-7", "Sorcerer Class Guide", "Learn how to play the Sorcerer Class", "Online", 35, 3),
  ("0-672-31738-3", "Warlock Class Guide", "Learn how to play the Warlock Class", "Online", 35, 3),
  ("0-672-31742-2", "Wizard Class Guide", "Learn how to play the Wizard Class", "Online", 35, 3),
  ("0-672-31747-6", "Artificer Class Guide", "Learn how to play the Artificier Class", "In-Person", 35, 3),
  ("0-672-31749-8", "Blood Hunter Class Guide", "Learn how to play the Druid Class", "In-Person", 20, 3),
  ("0-672-31753-1", "Intro to Subclasses", "Learn what subclasses are and how to play with a subclass", "In-Person", 25, 3),
  ("0-672-31756-7", "Dungeon Master Class Guide", "Learn how to be a Dungeon Master", "In-Person", 25, 3);
```

Courses Completed



Back to Dashboard

Your Completed Courses

1. Course Name: Dungeon Master Class Guide
Course Description: Learn how to be a Dungeon Master
Course ID: 0-672-31756-7
Seats: 25
Credits: 3

2. Course Name: Intro to Basics
Course Description: Learn the basics of Dungeons & Dragons.
Course ID: 0-672-31697-8
Seats: 50
Credits: 3

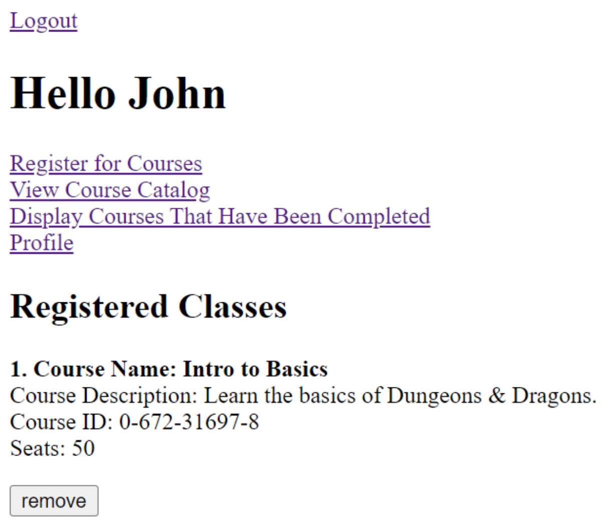
Total Credits: 6

Figure 5: Show user's completed courses

This SQL code defines and populates a table named "completedcourses" in a relational database. We use this code to allow a user to reference what classes they have completed. The code uses courseid and user_id to determine which course has been completed by which user.

```
CREATE TABLE `completedcourses` (
  `courseid` char(13) NOT NULL,
  `user_id` bigint(20) NOT NULL
);
INSERT INTO `completedcourses` (`courseid`, `user_id`) VALUES
('0-672-31753-1', 1),
('0-672-31698-9', 1),
('0-672-31703-5', 2),
('0-672-31749-8', 1),
('0-672-31697-8', 2),

('0-672-31698-9', 4043542408),

('0-672-31697-8', 4043542408),

('610610', 5713117084),

('0-672-31756-7', 5713117084);
```

Student in Course



Logout

Hello John

Register for Courses
View Course Catalog
Display Courses That Have Been Completed
Profile

Registered Classes

1. Course Name: Intro to Basics
Course Description: Learn the basics of Dungeons & Dragons.
Course ID: 0-672-31697-8
Seats: 50

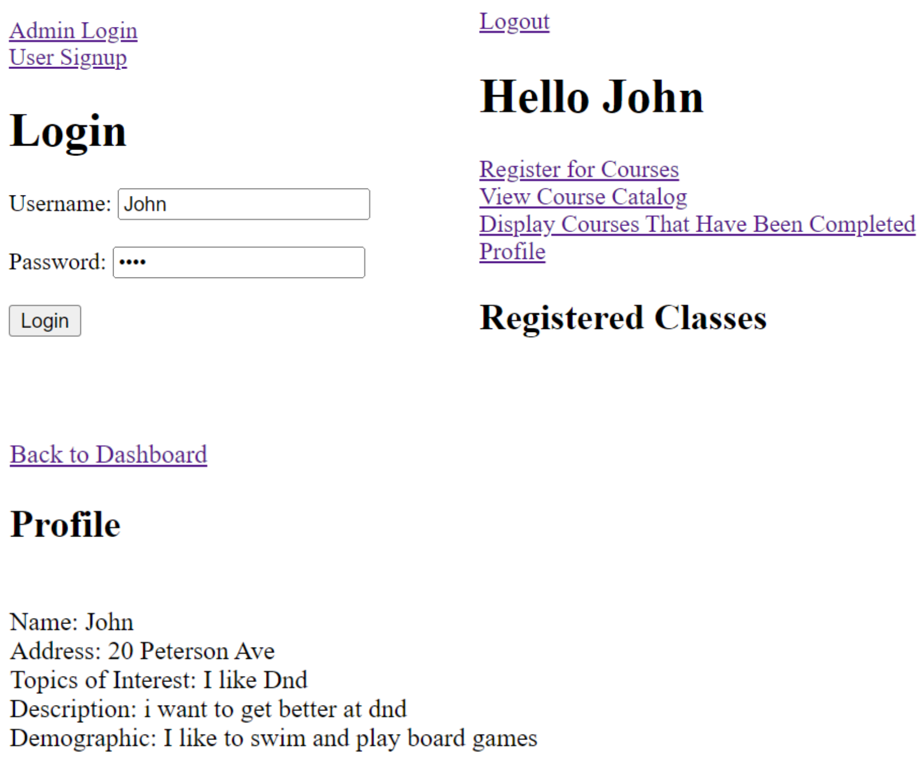
Figure 6: Student page - Show Courses That They Are In

This code defines a table named “studentincourse" in our relational database. This SQL code allows us to use the INSERT function, courseid, and user_id to determine what students are currently taking which courses.

```
CREATE TABLE `studentincourse` (

  `courseid` char(13) NOT NULL,

  `user_id` bigint(20) NOT NULL

)
```

```
INSERT INTO `studentincourse` (`courseid`, `user_id`) VALUES

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

('0-672-31697-8', 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 5713117084),

(", 2),

(", 2),

(", 2),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 2),
('', 2),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 5713117084),
('', 2),
('', 5713117084),
('', 5713117084),
('', 2),
('', 2),
('', 2),
('', 5713117084),
('', 2),
(", 5713117084),

(", 5713117084),

(", 5713117084);
```

User



[Admin Login](#)
[User Signup](#)

Login

Username: John

Password: ••••

Login

[Logout](#)

Hello John

[Register for Courses](#)
[View Course Catalog](#)
[Display Courses That Have Been Completed](#)
[Profile](#)

Registered Classes

[Back to Dashboard](#)

Profile

Name: John
Address: 20 Peterson Ave
Topics of Interest: I like Dnd
Description: i want to get better at dnd
Demographic: I like to swim and play board games

Figure 7: User Login, Dashboard, and Profile Page

This is the SQL code for users or clients. This code creates a table named ‘user’ in a relational database. This code is used as a reference for things like User_id, User_name, and Password which we need in order to secure our database as well as allow the creation and deletion of student profiles.

```
CREATE TABLE `user` (

  `id` bigint(20) NOT NULL,

  `user_id` bigint(20) NOT NULL,
  `user_name` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL,
  `Name` varchar(100) NOT NULL,
  `Address` varchar(200) NOT NULL,
  `Topics` varchar(200) NOT NULL,
  `Descrip` varchar(200) NOT NULL,
  `Demog` varchar(200) NOT NULL
);

INSERT INTO `user` (`id`, `user_id`, `user_name`, `password`, `Name`, `Address`, `Topics`,
`Descrip`, `Demog`) VALUES
(1, 1, 'jared', 'jared', '', '', '', '', ''),
(2, 2, 'dhruv', 'dhruv', '', '', '', '', ''),
(3, 6903561206, 'xavier', 'xavier', '', '', '', '', ''),
(4, 5508721796, 'stavros', 'stavros', '', '', '', '', ''),
(7, 5713117084, 'John', 'john', 'John', '20 Peterson Ave', 'I like Dnd', 'i want to get better at dnd', 'I
like to swim and play board games');

ALTER TABLE `user`
  ADD PRIMARY KEY (`id`);

ALTER TABLE `user`
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
```

COMMIT;

Data Model

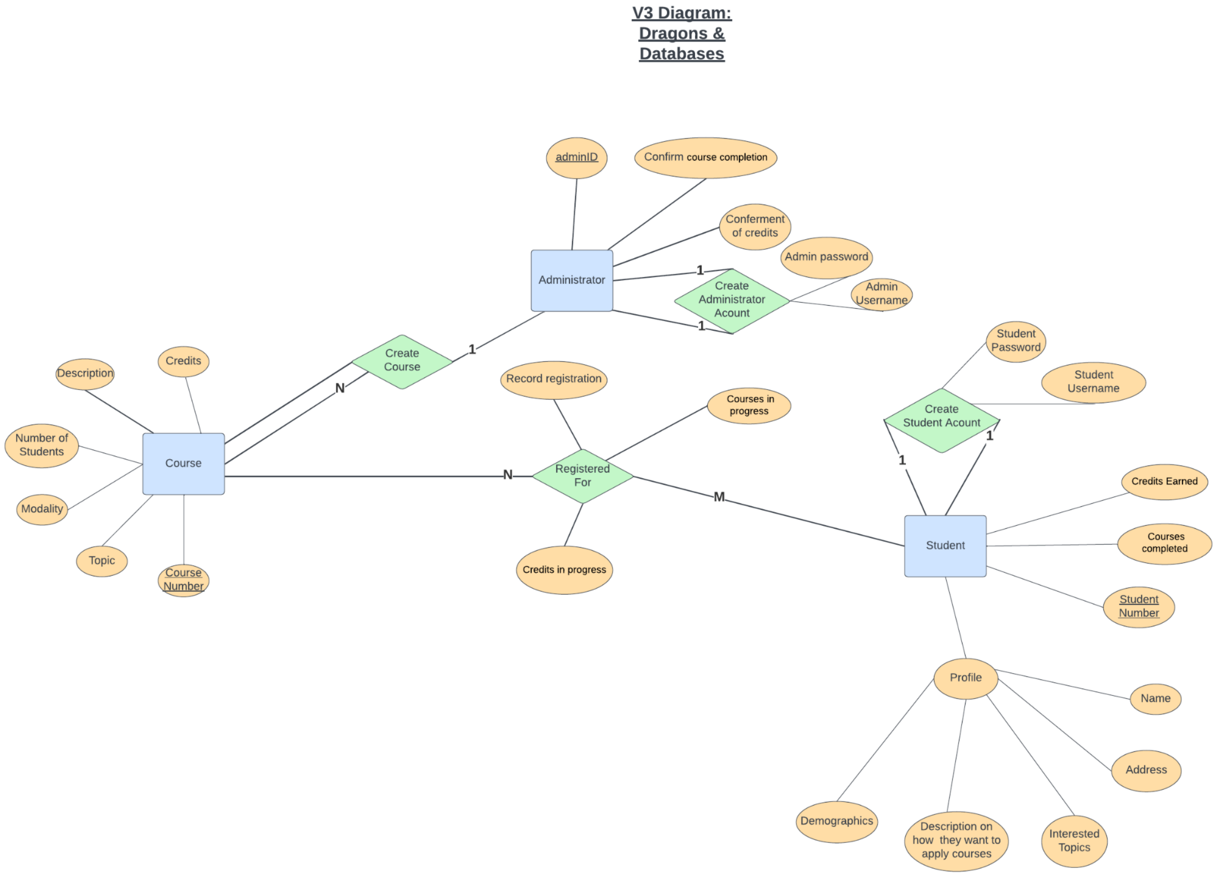


Figure 8: ER V3 Diagram of Dragons and Databases

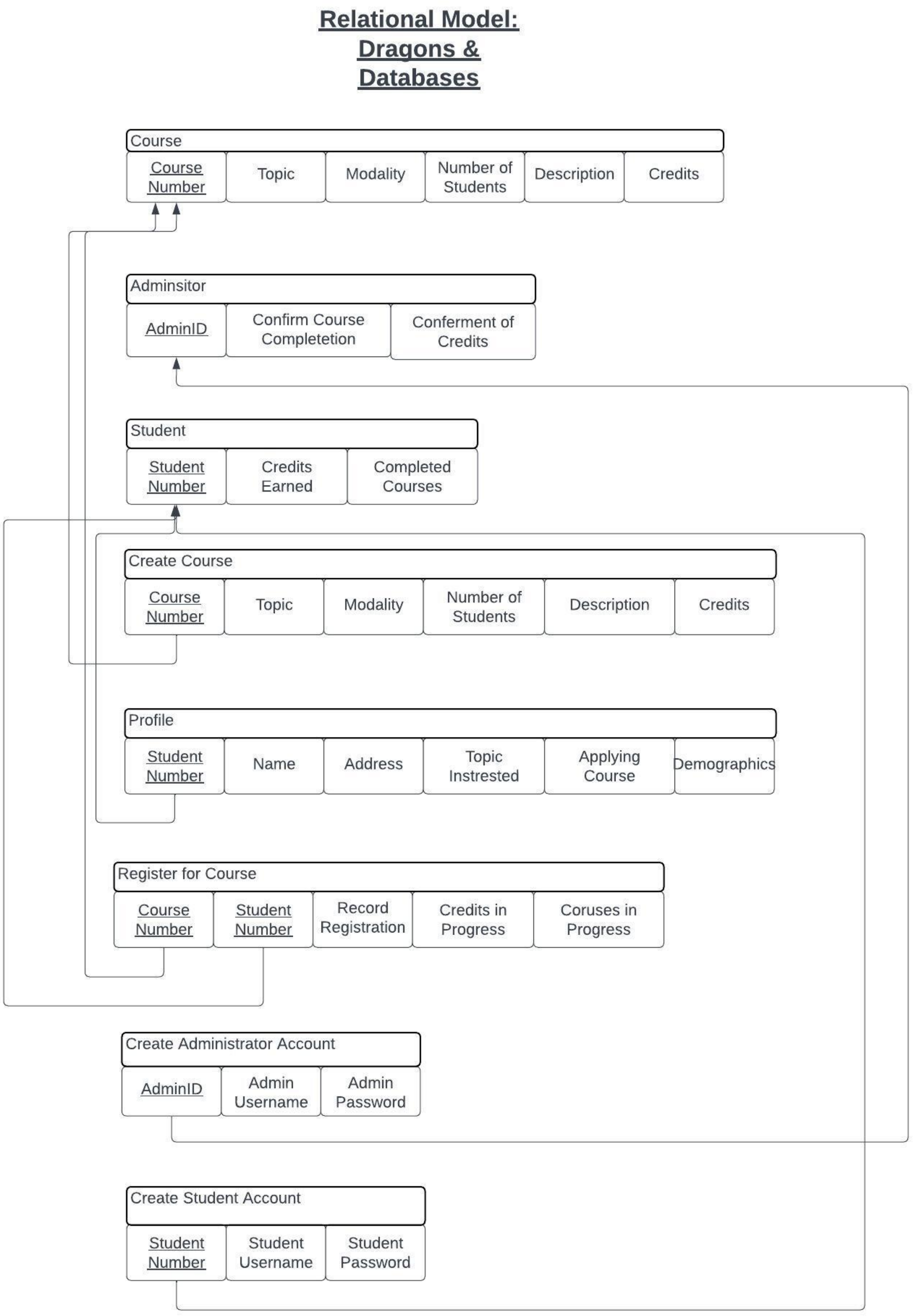
The figure above is the V3 ER diagram of the course registration system. The diagram meets these specific requirements that our employer was looking for:

- Courses have the following features:
  - Topic
  - Modality
  - Number of Students
  - Credits

    - Description
    - Course Number
- Students (Clients) have the following features:
    - Student Number
    - Credits Earned
    - Courses completed
    - Profile, which as the following attributes:
        - Name
        - Address
        - Interested Topics
        - Applying for Course
        - Demographics
- Administrator:
    - AdminID
    - Confirm Course Completion
    - Confirm Credits
- Relationships:
    - Administrator:
        - The Administrator has a 1-to-many (1:N) relationship with Courses. An Administrator is allowed to create a course.
        - The Administrator has a 1 to 1 (1:1) relationship with itself. An Administrator is allowed to create an Administrator account with an Admin username and Admin password.

            - Admin Password
            - Admin Username
    - Student:
        - The Student has a many-to-many (M: M) relationship with Courses. A student is allowed to register for courses.
            - Record Registration
            - Course in Progress
            - Credits in Progress
        - The student has a one-to-one (1: 1) relationship with itself. A Student is allowed to create a student account with a student username and student password
            - Student Password
            - Student Username

If we had more time we would implement a section for user feedback, the ability to create an employee, and to make the user interface more presentable. With time contractions that were imposed upon the team, we did not have enough time to implement the user feedback feature because it would interact with many different features, making the implementation difficult and time-consuming. When it comes to implementing employees into the system we did not implement it because we agreed that admin is the only thing we need for our system. After all, admin and employee would have the same privileges and since our system is only for registration we do not need an employee that is a teacher. Lastly, not an issue but we would have liked to make our user interface more visually appealing but due to time constraints we wanted to dedicate our resources to meeting the requirements first.



Relational Model: Dragons & Databases

Course

Course Number	Topic	Modality	Number of Students	Description	Credits

Adminsitor

AdminID	Confirm Course Completetion	Conferment of Credits

Student

Student Number	Credits Earned	Completed Courses

Create Course

Course Number	Topic	Modality	Number of Students	Description	Credits

Profile

Student Number	Name	Address	Topic Instrested	Applying Course	Demographics

Register for Course

Course Number	Student Number	Record Registration	Credits in Progress	Coruses in Progress

Create Administrator Account

AdminID	Admin Username	Admin Password

Create Student Account

Student Number	Student Username	Student Password

Figure 9: Relational Model of Dragons and Databases

The figure above is the Relation Model of the course registration system. The diagram shows how data is stored within a relational database. The Course class has a primary key, Course Number, and the attributes Topic, Modality, Number of Students, Description, and Credits. The Course Number is a foreign key in the table Create Course, since to create a course you must have a course identifier, followed by the course attributes of Topic, Modality, Number of Students, Description, and Credits. The Administrator table has the primary key AdminID, with the attributes Confirming Course Completion, and Conferment of Credits. The AdminID is a foreign key in Create Administrator Account to identify the administrator followed by the attributes Admin Username and Admin Password needed to create an account. The Student table has a primary key Student Number, and the attributes credits earned, and completed course. The table Profile, uses Student Number as a foreign key, to identify the student, followed by the attributes Name, Address, Topic Interested, Applying Course, and Demographic. The Student Number is a foreign key in the table Create Student Account, to identify the student, and then with the attributes Student Username and Student Password. Similarly, the Register for Course class, which is used by the student, uses the Course Number from table Course, and the Student Number from table Student, as a foreign keys to register a student to a class, and followed by the attributes Record Registration, Credits in Progress, and Courses in Progress.

Conclusion

In conclusion, the team created a database and a website for a local business that wanted to introduce new people to DND. We created 20 courses and used a combination of SQL, PHP, and HTML to create the website and the database. We created two different types of users and assigned each of them different privileges. Ultimately we achieved our goal of creating a user-friendly platform with easy course sign-ups and all of the necessary features to make the project functional. There were a few things we would have liked to add but we believe that our client will be satisfied with our final product.

Appendix (All Code)

index.php

This php file is responsible for the user login page, the first page the user will see when trying out our system. This php file is responsible for login users of the system and also has links for admin login and user signup. For a user to login to the system they must enter a username and password.

```
<?php
session_start();


$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');


// include("functions.php");


if($_SERVER['REQUEST_METHOD'] == "POST")
{
   $user_name = $_POST['user_name'];
   $password = $_POST['password'];


   if(!empty($user_name) && !is_numeric($user_name) && !empty($password))
   {
      $query = "select * from user where user_name = '$user_name' limit 1";


      $result = mysqli_query($db, $query);
```

```
        if($result && mysqli_num_rows($result) > 0)
        {
            $user_data = mysqli_fetch_assoc($result);

            if($user_data['password'] === $password)
            {
                $_SESSION['user_id'] = $user_data['user_id'];
                header("Location: dashboard.php");
                die;
            }
        }
        echo "You must enter the correct username or password";
    } else
    {
        echo "You must enter a valid username and password";
    }
}
?>

<html>
    <body>
        <br>
```

```
      <a href="admin_login.php">Admin Login</a>

      <br>

      <a href='signup.php'>User Signup</a>

      <h1> Login </h1>

      <form method="post">

        <label for="username">Username:</label>

        <input type="text" id="username" name="user_name" required><br><br>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password" required><br><br>

        <button type="submit"> Login </div>

      </form>

    </div>

    </body>

</html>
```

admin_login.php

This php file is responsible for the admin login page and it is responsible for letting admins login to their account and interact with the system. For the admin to login the admin needs to enter an admin username and admin password.

```
<?php
session_start();

$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');


if($_SERVER['REQUEST_METHOD'] == "POST")
{
    $admin_user_name = $_POST['admin_user_name'];
    $admin_password = $_POST['admin_password'];

    if(!empty($admin_user_name) && !is_numeric($admin_user_name) &&
!empty($admin_password))
    {
        $query = "select * from admin_user where admin_user_name = '$admin_user_name' limit
1";

        $result = mysqli_query($db, $query);

        if($result && mysqli_num_rows($result) > 0)
        {
            $user_data = mysqli_fetch_assoc($result);

            if($user_data['admin_password'] === $admin_password)
            {
                $_SESSION['userID'] = $user_data['userID'];
                header("Location: admin_dash.php");
                die;
            }
        }
        echo "You must enter the correct username or password";
    } else
    {
        echo "You must enter a valid username and password";
    }
}
?>

<html>
    <body>
        <br>
        <h1> Admin Login </h1>
        <form method="post">
```

```
            <label for="admin_username">Username:</label>

            <input type="text" id="admin_username" name="admin_user_name" required><br><br>

            <label for="password">Password:</label>

            <input type="password" id="admin_password" name="admin_password"

required><br><br>

            <button type="submit"> Login </div>

        </form>

    </div>

    </body>

</html>
```

signup.php

This php file is responsible for the user sign up page and is responsible for a client to make a user account so that they can interact with the system. The data needed to be entered for this php file is name, address, Interested Topics, How are you looking to apply what you learned?, Tell us a little about yourself, username, and password.

```
<?php
$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');


if($_SERVER['REQUEST_METHOD'] == "POST"){
    $user_name = $_POST['user_name'];
    $password = $_POST['password'];
    $Name = $_POST['Name'];
    $Address = $_POST['Address'];
    $Topics = $_POST['Topics'];
    $Descrip = $_POST['Descrip'];
    $Demog = $_POST['Demog'];



    if(!empty($user_name) & !is_numeric($user_name) & !empty($password)){
        $user_id = "";


        $len = rand(9, 10);
        for ($i=0; $i < $len; $i++) {
            $user_id .= rand(0,9);
        }

        $query = "insert into user (user_id, user_name, password, Name, Address, Topics, Descrip,
Demog) values ('$user_id','$user_name', '$password', '$Name', '$Address', '$Topics', '$Descrip',
'$Demog')";
        mysqli_query($db, $query);

        header("location: index.php");
        die;
    }
}


?>

<html>
    <body>
        <h1>Signup</h1>
        <br>
        <a href='index.php'>Back to Login</a>
        <br><br>
        <form method='post' action='signup.php'>
            <label for="Name"> Name:</label>

            <input type="text" id="Name" name="Name" required><br><br>


            <label for="Address"> Address:</label>

            <input type="text"id="Address" name="Address" required><br><br>


            <label for="Topics"> Interested Topics:</label>

            <input type="text"id="Topics" name="Topics" required><br><br>


            <label for="Descrip"> How are you looking to apply what you learned?:</label>

            <input type="text" id="Descrip" name="Descrip" required><br><br>


             <label for="Demog"> Tell us a little about yourself:</label>

            <input type="text" id="Demog" name="Demog" required><br><br>


            <label for="username">Username:</label>

            <input type="text" id="username" name="user_name" required><br><br>


            <label for="password">Password:</label>

            <input type="password" id="password" name="password" required><br><br>


            <button type="submit">Signup</button>

        </form>
```

```
  </body>
</html>
```

logout.php

This php file is responsible for login out the user from the system if requested too.

```
<?php
session_start();


if(isset($_SESSION['user_id']))
{
  unset($_SESSION['$user_id']);
}
header('Location: index.php');
die;


?>
```

admin_logout.php

This php file is responsible for login out the user from the system if requested too.

```
<?php
session_start();


if(isset($_SESSION['userID']))
{
   unset($_SESSION['$userID']);
}
header('Location: index.php');
die;


?>
```

admin_dash.php

This php file is the admin dashboard page and is responsible for having an admin page to show the admin which functions he is allowed to do and clickable links to these functions. The functions that the admin are allowed to do are add a new course and confirm courses have been completed.

```
<?php
session_start();


$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');


  // check login
  if(isset($_SESSION['userID'])) {
    $id = $_SESSION['userID'];
    $query = "select * from admin_user where userID = '$id' limit 1";


    $result = mysqli_query($db, $query);


    if($result && mysqli_num_rows($result) > 0) {
      $user_data = mysqli_fetch_assoc($result);
    }
  }
```

```
?>


<html>
  <body>
    <a href='admin_logout.php'>Logout</a>
    <h1>Hello <?php echo $user_data['admin_user_name']; ?></h1>
    <a href='newCourse.html'>Add A New Course</a><br>
    <a href='admin_complete_course.php'>Complete Active Course</a>
  </body>
</html>
```

newCourse.html

This html file is the new course page and is responsible for letting the admin enter data to the course database to add a new course. The data needed to be enter for the html file to work is course id, topic, description, modality, number of seats, and credits.

```
 <html>
<head>
  <title>Dragons and Databases - New Course Entry</title>
</head>


<body>
  <h1>Dragons and Databases - New Course Entry</h1>
  <form action="insertCourse.php" method="post">
    <table border="0">
```

```
<tr>
  <td>Course ID</td>
   <td><input type="text" name="courseID" maxlength="13" size="13"></td>
</tr>
<tr>
  <td>Course Name</td>
  <td> <input type="text" name="topic" maxlength="30" size="30"></td>
</tr>
<tr>
  <td>Course Description</td>
  <td> <input type="text" name="description" maxlength="60" size="30"></td>
</tr>
<tr>
  <td>Modality</td>
  <td><input type="text" name="modality" maxlength="7" size="7"></td>
</tr>
<tr>
  <td>Number of Seats</td>
  <td><input type="text" name="numberStudents" maxlength="7" size="7"></td>
</tr>
<tr>
  <td>Credits</td>
  <td><input type="text" name="credits" maxlength="7" size="7"></td>
```

```
      </tr>
      <tr>
        <td colspan="2"><input type="submit" value="Register"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

insertCourse.php

This php file is responsible for confirming to the admin that the admin has successfully imputed a new course in the course database. If not successful the php file will tell the admin that it wasn't successful.

```
<html>
<head>
  <title>Dragons and Databases Add A Course</title>
</head>
<body>
<h1>Dungeons And Databases Add A Course Result</h1>
<?php
  // create short variable names
  $courseID=$_POST['courseID'];
  $topic=$_POST['topic'];
  $description=$_POST['description'];
```

```
$modality=$_POST['modality'];

$numberStudents=$_POST['numberStudents'];

$credits=$_POST['credits'];


if (!$courseID || !$topic || !$description || !$modality) {

   echo "You have not entered all the required details.<br />"

        ."Please go back and try again.";

   exit;

}


if (!get_magic_quotes_gpc()) {

  $courseID = addslashes($courseID);

  $topic = addslashes($topic);

  $description = addslashes($description);

  $modality = addslashes($modality);

  $numberStudents = doubleval($numberStudents);

  $credits = doubleval($credits);

}


@$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731', 'warnerx3_DnDFinal');


if (mysqli_connect_errno()) {

   echo "Error: Could not connect to database.  Please try again later.";
```

```
    exit;
  }


  $query = "insert into courses values
          ('".$courseID."', '".$topic."', '".$description."', '".$modality."', '".$numberStudents."',
'".$credits."')";
  $result = $db->query($query);


  if ($result) {
     echo  $db->affected_rows." Course inserted into database.";
  } else {
           echo "An error has occurred.  The item was not added.";
  }


  $db->close();
?>
</body>
</html>
```

admin_complete_course.php

This php file is the complete courses page and is responsible for letting the admin to confirm that a student(user) has completed a course. Allowing the user to see the courses that they have completed and the amount of credits they have as well.

```
<?php
```

```
session_start();
$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');


if($_SERVER['REQUEST_METHOD'] == "POST") {
   $courseid = $_POST['courseID'];


   $query =
   "insert into completedcourses
   select * from studentincourse
   where studentincourse.courseid = '$courseid'";
   mysqli_query($db, $query);


   $query = "delete from studentincourse where studentincourse.courseid='$courseid'";
   mysqli_query($db, $query);






   // header("Location: dashboard.php");
   // die;
}


?>
```

```
<html>
    <body>
        <a href='admin_dash.php'>Back to Admin Dashboard</a>

        <h1>Current Active Courses</h1>
        <?php
        $query = "select DISTINCT c.courseID, topic from courses c
        join studentincourse s on c.courseID = s.courseid";

        $result = $db->query($query);

        $num_results = $result->num_rows;

        for($i = 0; $i < $num_results; $i++){
            $row = $result->fetch_assoc();
            echo "<p><strong>".($i+1).". Course Name: ";
            echo htmlspecialchars(stripslashes($row['topic']));
            echo '<form method="POST" action="admin_complete_course.php"><input
type="hidden" name="courseID" value="'.$row['courseID'].'"><input type="submit"
value="Complete Course" name="submit"></form>';
        }
    ?>
  </body>
</html>
```

dashboard.php

This php file is the user dashboard page and is responsible for a user(student) page that informs the user of what functions they are allowed to do and clickable links to these functions. These functions are register for a class, display courses that have been completed, and profile. The user dashboard page also shows the user which classes they have registered for.

```
<?php
session_start();


$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');


  // check login
  if(isset($_SESSION['user_id'])) {
    $id = $_SESSION['user_id'];
    $query = "select * from user where user_id = '$id' limit 1";


    $result = mysqli_query($db, $query);


    if($result && mysqli_num_rows($result) > 0) {
      $user_data = mysqli_fetch_assoc($result);
    }
  }

  if($_SERVER['REQUEST_METHOD'] == "POST")
  {
    $courseid = $_POST['courseID'];
    $user_id = $_SESSION['user_id'];
    // echo $courseid;
    // echo $user_id;

    $query = "delete from studentincourse where courseid = '$courseid' and user_id = '$user_id'";
    mysqli_query($db, $query);

    header("Location: dashboard.php");
    die;
  }

?>

<html>
  <body>
    <a href='logout.php'>Logout</a>
    <h1>Hello <?php echo $user_data['user_name']; ?></h1>
```

```
<a href='register.php'>Register for Class</a><br>
<a href='user_completed.php'>Display Courses That Have Been Completed</a><br>
<a href='profile.php'>Profile</a><br>
<h2>Registered Classes</h2>
<?php
  $id = $user_data['user_id'];


  $query = "select *
  from courses c
  join studentincourse s on c.courseID = s.courseid
  join user u on u.user_id = s.user_id
  where u.user_id = $id";

  $result = $db->query($query);

  $num_results = $result->num_rows;

  for ($i=0; $i <$num_results; $i++) {
    $row = $result->fetch_assoc();
    echo "<p><strong>".($i+1).". Course Name: ";
    echo htmlspecialchars(stripslashes($row['topic']));
```

```
                echo "</strong><br />Course Description: ";
                echo stripslashes($row['description']);
                echo "<br />Course ID: ";
                echo stripslashes($row['courseID']);
                echo "<br />Seats: ";
                echo stripslashes($row['numberStudents']);
                echo "</p>";
                echo '<form method="POST" action="dashboard.php"><input type="hidden" name="courseID" value="'.$row['courseID'].'"><input type="submit" value="remove" name="submit"></form>';

            }

            $result->free();
            $db->close();
        ?>
    </body>
</html>
```

register.php

This php file is the register course page and is responsible for the page that allows the user to view courses and select which course they want to register for. Which will then direct them back to the user dashboard page.

```
<?php
session_start();


$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');
// echo $_SESSION['user_id'];


   // check login
   if(isset($_SESSION['user_id'])) {
      $id = $_SESSION['user_id'];
      $query = "select * from user where user_id = '$id' limit 1";


      $result = mysqli_query($db, $query);


      if($result && mysqli_num_rows($result) > 0) {
         $user_data = mysqli_fetch_assoc($result);
         // return $user_data;
      }
   }
```

```
  if($_SERVER['REQUEST_METHOD'] == "POST")
  {
    $courseid = $_POST['courseID'];
    $user_id = $_SESSION['user_id'];
    // echo "courseid: ".$courseid;
    // echo "userid:".$user_id;

    $query = "insert into studentincourse (courseid, user_id) values ('$courseid', '$user_id')";
    mysqli_query($db, $query);

    header("Location: dashboard.php");
    die;
  }

?>

<html>
  <body>
    <h2>Courses</h2>
    <br>

    <?php
```

```
        $query = "select * from courses ";
        $result = $db->query($query);

        $num_results = $result->num_rows;



        for ($i=0; $i <$num_results; $i++)
        {
           $row = $result->fetch_assoc();
           echo "<p><strong>".($i+1).". Course Name: ";
           echo htmlspecialchars(stripslashes($row['topic']));
           echo "</strong><br />Course Description: ";
           echo stripslashes($row['description']);
           echo "<br />Course ID: ";
           echo stripslashes($row['courseID']);
           echo "<br />Seats: ";
           echo stripslashes($row['numberStudents']);
           echo "</p>";
           echo '<form method="POST" action="register.php"><input type="hidden"
name="courseID" value="'.$row['courseID'].'"><input type="submit" value="register"
name="submit"></form>';
```

```
            }
        ?>


    </body>
</html>
```

searchCourse.html

This html file is the search course page and is responsible for letting the user search up a course based on a selection of course id, description, and course name by user input.

```
<html>
<head>
  <title>Dragons And Databases Catalog Search</title>
</head>


<body>
  <h1>Dragons And Databases Catalog Search</h1>


  <form action="searchPage.php" method="post">
    Choose Search Type:<br />
    <select name="searchtype">
      <option value="topic">Course Name
      <option value="description">Course Description
      <option value="courseID">Course ID
    </select>
    <br />
    Enter Search Term:<br />
    <input name="searchterm" type="text" size="40">
    <br />
    <input type="submit" name="submit" value="Search">
  </form>

</body>
</html>
```

searchPage.php

This php file is the search page and is responsible for letting the user see what courses they have searched up from the searchCourse.html file.

```
<?php
@$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731', 'warnerx3_DnDFinal');
if (mysqli_connect_errno()) {
    echo 'Error: Could not connect to database.  Please try again later.';
    exit;
  }
?>

<html>
<head>
  <title>Dragons and Databases Catalog Results</title>
```

```
</head>
<body>
<h1>Dragonss and Databases Catalog Results</h1>
<?php
  // create short variable names
  $searchtype=$_POST['searchtype'];
  $searchterm=trim($_POST['searchterm']);

  if (!$searchtype || !$searchterm) {
     echo 'You have not entered search details.  Please go back and try again.';
     exit;
  }

  if (!get_magic_quotes_gpc()){
    $searchtype = addslashes($searchtype);
    $searchterm = addslashes($searchterm);
  }



  $query = "select * from courses where ".$searchtype." like '%".$searchterm."%'";
  $result = $db->query($query);
  $num_results = $result->num_rows;


  echo "<p>Number of Courses found: ".$num_results."</p>";


  for ($i=0; $i <$num_results; $i++) {
     $row = $result->fetch_assoc();
     echo "<p><strong>".($i+1).". Course: ";
     echo htmlspecialchars(stripslashes($row['topic']));
     echo "</strong><br />Description: ";
     echo stripslashes($row['description']);
     echo "<br />Course ID: ";
     echo stripslashes($row['courseID']);
     echo "<br />Seats: ";
     echo stripslashes($row['numberStudents']);
     echo "</p>";
  }


  $result->free();
  $db->close();


?>
</body>
</html>
```

user_completed.php

This php file is the user course completed page and is responsible for the page that informs the user which courses they have completed and how many credits they have earned as well.

```
<?php
   session_start();


   $db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');


   // check login
   if(isset($_SESSION['user_id'])) {
      $id = $_SESSION['user_id'];
      $query = "select * from user where user_id = '$id' limit 1";


      $result = mysqli_query($db, $query);


      if($result && mysqli_num_rows($result) > 0) {
         $user_data = mysqli_fetch_assoc($result);
      }
   }
```

```
?>

<html>
    <body>
        <a href='dashboard.php'>Back to Dashboard</a>
        <h1>Your Completed Courses</h1>



        <?php
            $id = $user_data['user_id'];
            $creditcumulative = 0;

            // $query = "select DISTINCT c.courseID, topic from courses c join completedcourses s
on c.courseID = s.courseid";
            $query = "select * from courses c join completedcourses cc on cc.courseid = c.courseID
join user u on u.user_id = cc.user_id where u.user_id = $id";

            $result = $db->query($query);

            $num_results = $result->num_rows;

            for($i=0; $i <$num_results; $i++) {
                $row = $result->fetch_assoc();

                $creditcumulative = $creditcumulative + $row['credits'];

                echo "<p><strong>".($i+1).". Course Name: ";

                echo htmlspecialchars(stripslashes($row['topic']));

                echo "</strong><br />Course Description: ";

                echo stripslashes($row['description']);

                echo "<br />Course ID: ";

                echo stripslashes($row['courseID']);

                echo "<br />Seats: ";

                echo stripslashes($row['numberStudents']);

                echo "<br />Credits: ";

                echo stripslashes($row['credits']);

                echo "</p>";

            }

            echo "<p>Total Credits: ";

            echo $creditcumulative;

            echo "</p>"

        ?>

    </body>

</html>
```

profile.php

This php file is the user profile page and is responsible for informing the user of their profile details. The profile page will include the following information to be displayed for the user, name, address, topics of interest, description, and demographic

```
<?php
session_start();


$db = new mysqli('localhost', 'warnerx3_binary', 'bionicle731','warnerx3_DnDFinal');
// echo $_SESSION['user_id'];


  // check login
  if(isset($_SESSION['user_id'])) {
    $id = $_SESSION['user_id'];
    $query = "select * from user where user_id = '$id' limit 1";


    $result = mysqli_query($db, $query);


    if($result && mysqli_num_rows($result) > 0) {
      $user_data = mysqli_fetch_assoc($result);
      // return $user_data;
    }
  }
```

```
    if($_SERVER['REQUEST_METHOD'] == "POST")
    {
        $courseid = $_POST['courseID'];
        $user_id = $_SESSION['user_id'];
        // echo "courseid: ".$courseid;
        // echo "userid:".$user_id;

        $query = "insert into studentincourse (courseid, user_id) values ('$courseid', '$user_id')";
        mysqli_query($db, $query);

        header("Location: dashboard.php");
        die;
    }

?>

<html>
    <body>
        <a href='dashboard.php'>Back to Dashboard</a>
        <h2>Profile</h2>
        <br>
        <?php
            $query = "select * from user where user_id = $id";
            $result = $db->query($query);

            $num_results = $result->num_rows;

            for ($i=0; $i <$num_results; $i++) {
                $row = $result->fetch_assoc();
                echo " Name: ";
                echo stripslashes($row['Name']);
                 echo "</strong><br />Address: ";
                echo stripslashes($row['Address']);
                echo "<br />Topics of Interest: ";
                echo stripslashes($row['Topics']);
                echo "<br />Description: ";
                echo stripslashes($row['Descrip']);
                echo "<br />Demographic: ";
                echo stripslashes($row['Demog']);
                echo "</p>";
            }
        ?>
    </body>
</html>
```