ARCHERY DATABASE DESIGNING PROJECT

Tran Duc Anh Dang - 103995439
Eloise Grant - 103947870
Luan Nguyen - 103812143

Introduction

- Target archery requires precise measurements and scores reflecting the archer's skill.
- The sport requires accurate shooting at targets set at varied distances.
- Manual maintenance and organization of score data can be challenging and prone to errors.
- The presentation introduces the project - the Club Database for Archery Score Recording.
- The project aims to systematize and streamline the archery score recording process for clubs.

https://mercury.swin.edu.au/cos20031/s103995439/site/index.php

Project Background

- The scoring system in target archery is complex, with various factors influencing the final score. Participants shoot arrows at a target set at a certain distance in a sequence called an 'end'. An archer's performance is recorded as a 'score', which is the culmination of several 'ends'.
- Competitions incorporate rounds comprising different ranges and ends. The round definitions are unique and are often named after Australian cities. Archers are classified based on age, gender, and equipment type, defining their competition category.
- Our primary stakeholders are the archers themselves, who need to monitor their scores over time and the club recorder, responsible for ensuring the accurate entry of scores.
- This project aims to create a comprehensive database for maintaining individual and competition scores, facilitating easy access, record updating, and overall score management. This will provide archers and recorders with a smooth and effective method to manage archery scores, thereby enhancing their experience in the sport.

Mission statement & Objectives

Mission Statement

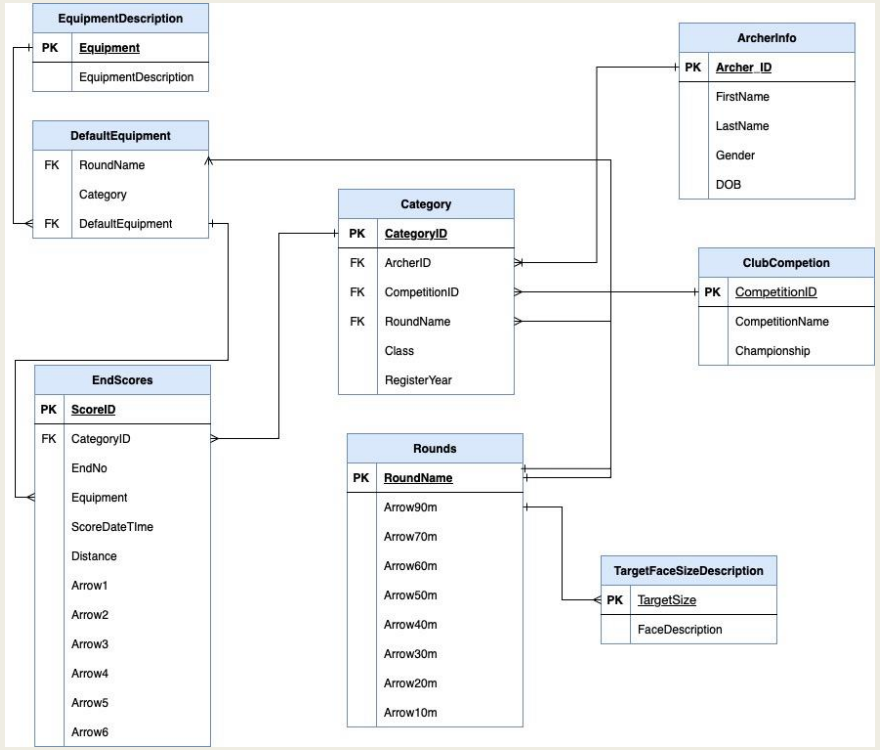
- To revolutionize archery score management by developing an intuitive and comprehensive Club Database for Archery Score Recording, enhancing the sporting experience for archers and simplifying the administration process for club recorders.

Objectives

- **User-centric Design:** Develop a database that is easy to use and understand for archers and recorders, enabling them to enter and retrieve scores with minimal effort.
- **Data Accuracy and Integrity:** Ensure accuracy and integrity in score recording by implementing robust data validation and control mechanisms.
- **Personalization:** Enable archers to view their individual scores, track their progress over time, and set personalized targets based on their historical performance.
- **Real-Time Updates:** Facilitate real-time score updates and allow archers to record scores using hand-held devices, ensuring quick and efficient data entry.
- **Historical Data Management:** Incorporate time-dependent changes effectively without invalidating past competitions, maintaining a clear and precise history of scores and rounds.

Database Design – ER Diagram

- The Archer Database is a holistic system for managing archery-related data.
- It comprises entities such as ArcherInfo, Rounds, TargetFaceSize, Category, ClubCompetitions, DefaultEquipment, EquipmentDescription, and EndScores.
- The database enables efficient data retrieval and analysis.
- It supports fair competition management and performance evaluation.
- The system serves as a valuable resource for archery enthusiasts.
- It provides a centralized platform for managing and analyzing archery data.



```sql
SELECT
    ClubCompetition.CompetitionName,
    Rounds.RoundName,
    ArcherInfo.FirstName,
    ArcherInfo.LastName,
    SUM(EndScores.Arrow1 + EndScores.Arrow2 + EndScores.Arrow3 + EndScores.Arrow4 + IFNULL(EndScores.Arrow5, 0
FROM
    ClubCompetition
JOIN
    Category ON ClubCompetition.CompetitionID = Category.CompetitionID
JOIN
    ArcherInfo ON Category.ArcherID = ArcherInfo.ArcherID
JOIN
    EndScores ON Category.CategoryID = EndScores.CategoryID
JOIN
    Rounds ON Category.RoundName = Rounds.RoundName
GROUP BY
    ClubCompetition.CompetitionName,
    Rounds.RoundName,
    ArcherInfo.FirstName,
    ArcherInfo.LastName
ORDER BY
    ClubCompetition.CompetitionName,
    Rounds.RoundName,
    TotalScore DESC;

```

CompetitionName	RoundName	FirstName	LastName	TotalScore
Australian Target Championships	AA	Christoph	Cowie	346

SQL – CHECKING WINNER FOR EACH ROUND IN A COMPETITION

rounds.php

```
261  SELECT
262      ArcherInfo.FirstName,
263      ArcherInfo.LastName,
264      EndScores.Distance,
265      EndScores.EndNo,
266      EndScores.Arrow1,
267      EndScores.Arrow2,
268      EndScores.Arrow3,
269      EndScores.Arrow4,
270      EndScores.Arrow5,
271      EndScores.Arrow6
272  FROM
273      ArcherInfo
274  JOIN
275      Category ON ArcherInfo.ArcherID = Category.ArcherID
276  JOIN
277      EndScores ON Category.CategoryID = EndScores.CategoryID
278  WHERE
279      ArcherInfo.FirstName = 'Christoph' AND ArcherInfo.LastName = 'Cowie';
```

Snipped

FirstName	LastName	Distance	EndNo	Arrow1	Arrow2	Arrow3	Arrow4	Arrow5	Arrow6
Christoph	Cowie	20m	1	10	9	8	7	NULL	NULL
Christoph	Cowie	20m	2	10	9	8	7	NULL	NULL
Christoph	Cowie	20m	1	10	9	8	7	NULL	NULL
Christoph	Cowie	20m	2	10	9	8	7	NULL	NULL
Christoph	Cowie	20m	1	10	9	8	7	NULL	NULL
Christoph	Cowie	20m	2	10	9	8	7	NULL	NULL
Christoph	Cowie	20m	1	10	9	9	8	NULL	NULL
Christoph	Cowie	30m	2	10	10	9	8	NULL	NULL
Christoph	Cowie	40m	3	9	9	9	8	NULL	NULL
Christoph	Cowie	50m	4	10	8	9	7	NULL	NULL

SQL – LOOKUP AN ARCHER'S SCORES

FirstName	LastName	Class	Equipment	RegisterYear	CompetitionName
Christoph	Cowie	Male Open	B	2023	Australian Target Championships
Christoph	Cowie	Male Open	B	2023	Australian Target Championships
Christoph	Cowie	Male Open	B	2023	Australian Target Championships
Luan	Nguyen	Male Open	B	2023	Australian Target Championships
Shirlee	Higgan	50+ Male	B	2023	Australian Target Championships
Benni	Deakes	Under 18 Female	B	2023	Australian Target Championships
Shirlee	Higgan	50+ Female	L	2023	Australian Target Championships
Luan	Bede	Female Open	R	2023	Australian Target Championships
Christoph	Cowie	Male Open	L	2023	Australian Target Championships
Christoph	Cowie	Male Open	L	2023	Australian Target Championships
Christoph	Cowie	Male Open	R	2023	Australian Indoor Championships
Christoph	Cowie	Male Open	L	2023	Australian Target Championships
Corbie	MacCulloch	Male Open	C	2023	Australian Indoor Championships
Benni	Deakes	Male Open	L	2023	Australian 3D Championships
Christoph	Cowie	Male Open	R	2023	Australian 3D Championships
Christoph	Cowie	Class 1	Equip 1	2023	Australian Target Championships
Nettie	Kerne	Class 1	Equip 1	2023	Australian Target Championships
Finley	Clawe	Class 1	Equip 1	2023	Australian Target Championships

rounds.php

```
261  SELECT
262      ArcherInfo.FirstName,
263      ArcherInfo.LastName,
264      Category.Class,
265      Category.Equipment,
266      Category.RegisterYear,
267      ClubCompetition.CompetitionName
268  FROM Category
269  INNER JOIN ArcherInfo ON Category.ArcherID = ArcherInfo.ArcherID
270  INNER JOIN ClubCompetition ON Category.CompetitionID = ClubCompetition.CompetitionID;
```

Snipped

SQL – CHECKING ASSIGNED CATEGORY

rounds.php

```
261  SELECT
262      ArcherInfo.FirstName,
263      ArcherInfo.LastName,
264      MAX(EndScores.Arrow1 + EndScores.Arrow2 + EndScores.Arrow3 + EndScores.Arrow4) AS PersonalBest
265  FROM EndScores
266  INNER JOIN Category ON EndScores.CategoryID = Category.CategoryID
267  INNER JOIN ArcherInfo ON Category.ArcherID = ArcherInfo.ArcherID
268  GROUP BY ArcherInfo.ArcherID
269  ORDER BY PersonalBest DESC;
```

Snipped

FirstName	LastName	PersonalBest
Christoph	Cowie	37

SQL – CHECKING PERSONAL BEST

rounds.php

```
261  SELECT
262      ArcherInfo.FirstName,
263      ArcherInfo.LastName,
264      SUM(EndScores.Arrow1 + EndScores.Arrow2 + EndScores.Arrow3 + EndScores.Arrow4) AS OverallScore
265  FROM EndScores
266  INNER JOIN Category ON EndScores.CategoryID = Category.CategoryID
267  INNER JOIN ArcherInfo ON Category.ArcherID = ArcherInfo.ArcherID
268  GROUP BY ArcherInfo.ArcherID
269  ORDER BY OverallScore DESC;
```

Snipped

FirstName	LastName	OverallScore
Christoph	Cowie	346

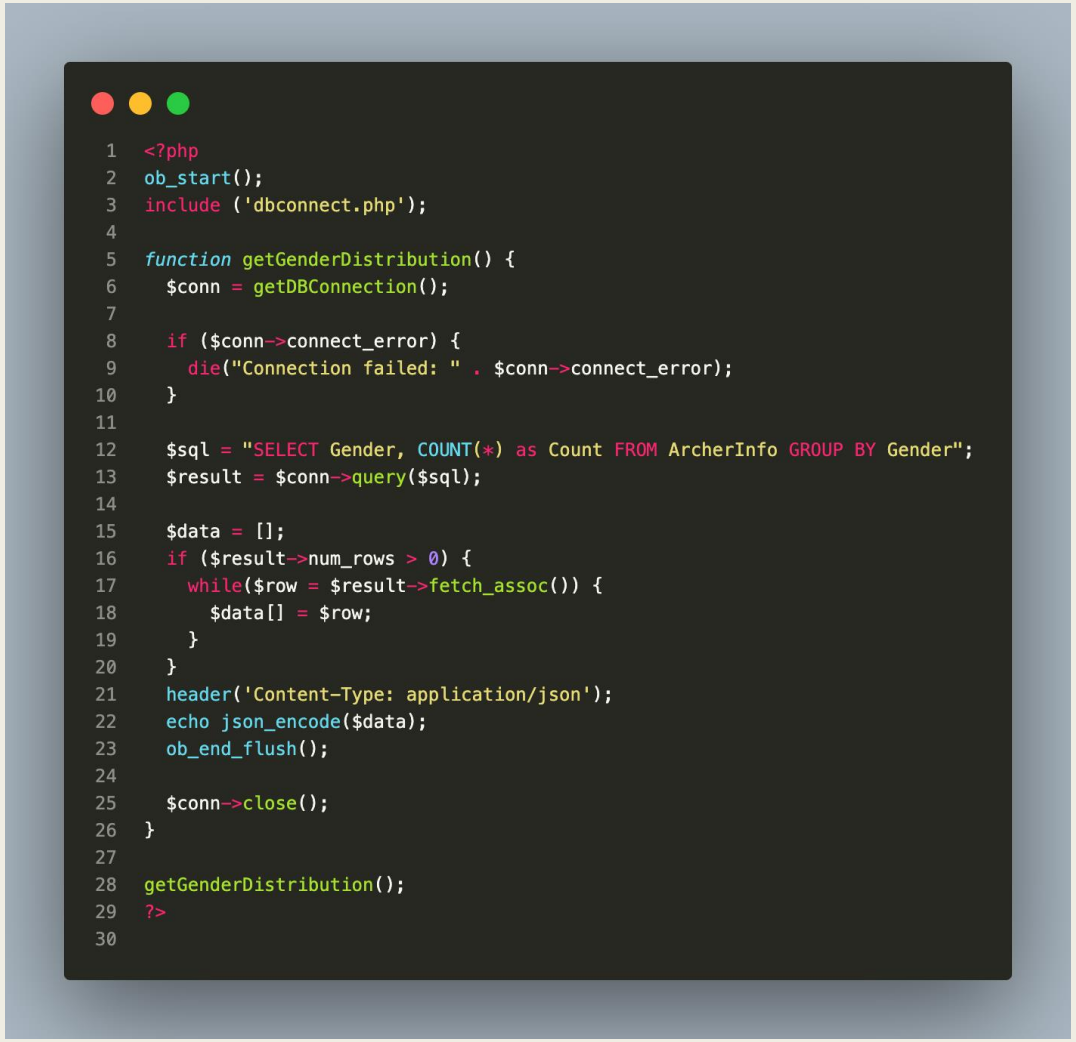
SQL – CHECKING OVERALL SCORE

```sql
-- Drop table
DROP TABLE IF EXISTS EndScores;
DROP TABLE IF EXISTS Category;
DROP TABLE IF EXISTS ClubCompetition;
DROP TABLE IF EXISTS DefaultEquipment;
DROP TABLE IF EXISTS EquipmentDescription;
DROP TABLE IF EXISTS TargetFaceSizeDescription;
DROP TABLE IF EXISTS Rounds;
DROP TABLE IF EXISTS ArcherInfo;

-- TABLES
-- ArcherInfo_Table
CREATE TABLE ArcherInfo (
    ArcherID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(255),
    LastName VARCHAR(255),
    Gender ENUM('M', 'F'),
    DOB DATE
) AUTO_INCREMENT=10000;

-- Round_Table
CREATE TABLE Rounds(
    RoundName VARCHAR(255) NOT NULL,
    Arrows90m CHAR(10) DEFAULT '0',
    Arrows70m CHAR(10) DEFAULT '0',
    Arrows60m CHAR(10) DEFAULT '0',
    Arrows50m CHAR(10) DEFAULT '0',
    Arrows40m CHAR(10) DEFAULT '0',
    Arrows30m CHAR(10) DEFAULT '0',
    Arrows20m CHAR(10) DEFAULT '0',
    Arrows10m CHAR(10) DEFAULT '0',
    PRIMARY KEY (RoundName)
);
```

Database Design – Tables

- Our database design is a comprehensive approach to efficiently store, retrieve and manage archery score data. The design comprises of eight tables to capture detailed information about the archers, rounds, equipment, competitions, and arrow scores.

1. ArcherInfo Table: Stores detailed information about each archer, including their unique ArcherID, First and Last names, Gender, and Date of Birth.
2. Rounds Table: Contains all-round definitions with a unique name for each round and the number of arrows shot at each distance.
3. TargetFaceSizeDescription Table: This table stores the descriptions for different target face sizes.
4. EquipmentDescription Table: Captures detailed descriptions for each type of archery equipment.
5. DefaultEquipment Table: Contains default equipment used for each round in each category. This table references the Rounds table.
6. ClubCompetition Table: Records information about club competitions, including a unique CompetitionID and the Championship if applicable.
7. Category Table: Stores data regarding archer's category, including the ArcherID, CompetitionID, and RoundName. The data in this table relates to the ArcherInfo, ClubCompetition, and Rounds tables.
8. EndScores Table: Captures detailed scoring data, including a unique ScoreID for each end, the CategoryID, the Equipment used, the shooting Distance, and scores for each of the six arrows. This table references the Category table.



```php
<?php
ob_start();
include ('dbconnect.php');

function getGenderDistribution() {
  $conn = getDBConnection();

  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
  }

  $sql = "SELECT Gender, COUNT(*) as Count FROM ArcherInfo GROUP BY Gender";
  $result = $conn->query($sql);

  $data = [];
  if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
      $data[] = $row;
    }
  }
  header('Content-Type: application/json');
  echo json_encode($data);
  ob_end_flush();

  $conn->close();
}

getGenderDistribution();
?>
```

MAJOR – A.I/DATASCIENCE

- SQL code aims to calculate and display gender distribution among archers, extracting gender data from the 'ArcherInfo' table, counting gender occurrences, and grouping them.
- In the PHP script, 'getGenderDistribution()' function is defined to connect with the database and execute the SQL query, outputting an error message and terminating if a connection error occurs.
- SQL query - "SELECT Gender, COUNT(*) as Count FROM ArcherInfo GROUP BY Gender" - retrieves gender and corresponding count from the 'ArcherInfo' table, grouping the result-set by the 'Gender' column.
- Post query execution, the result is fetched as an associative array and encoded in JSON format for further use, such as data visualization or web presentation.
- The function offers a simple way to understand the club's gender balance, which could inform initiatives promoting gender equality or diversification in club activities.

MAJOR – A.I/DATASCIENCE

- SQL query calculates and illustrates distribution of archers by classes, extracting class data from the 'Category' table and counting unique class occurrences.
- PHP function 'getClassDistribution()' initiates a database connection and executes the SQL query; any connection error triggers an error message and halts the script.
- SQL statement - "SELECT Class, COUNT(*) as Count FROM Category GROUP BY Class" - fetches class name and corresponding count, grouping result-set by the 'Class' column.
- After executing the query, results are fetched as an associative array and converted into JSON format for ease of interpretation and further use.
- Function aids understanding of archer distribution across classes, assisting in decision-making processes related to class-specific training, resource allocation, and event planning.

getClassDistribution.php

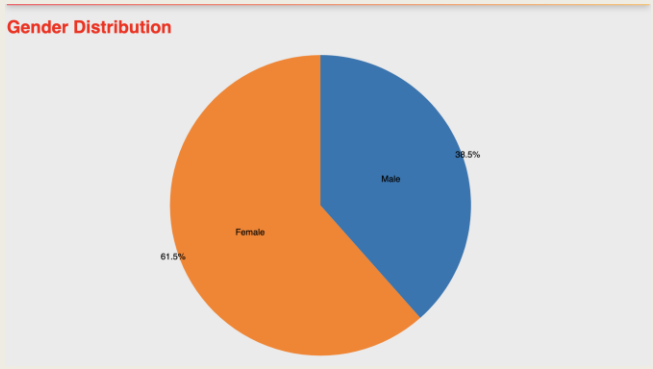
```php
<?php
ob_start();
include ('dbconnect.php');

function getClassDistribution() {
  $conn = getDBConnection();

  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
  }

  $sql = "SELECT Class, COUNT(*) as Count FROM Category GROUP BY Class";
  $result = $conn->query($sql);

  $data = [];
  if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
      $data[] = $row;
    }
  }
  header('Content-Type: application/json');
  echo json_encode($data);
  ob_end_flush();

  $conn->close();
}

getClassDistribution();
?>
```

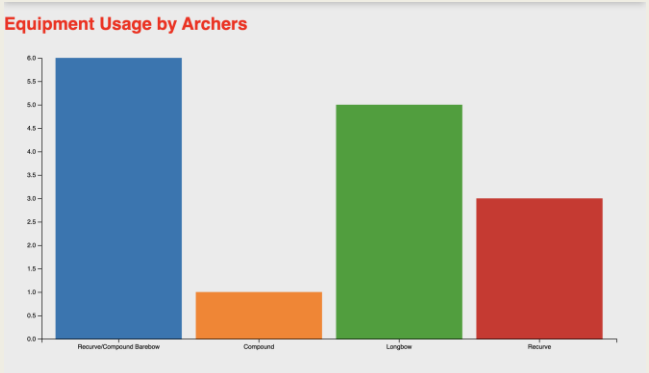
Snipped

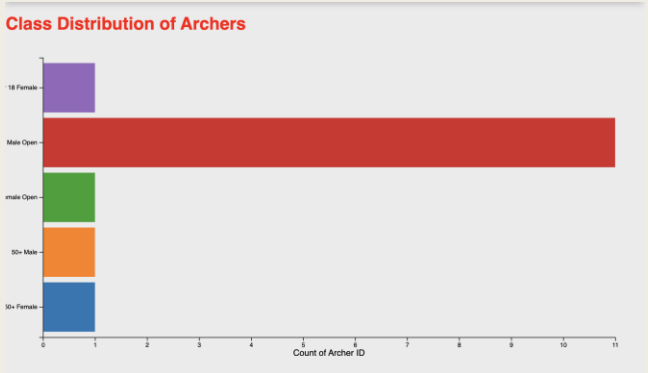
```php
<?php
ob_start();
include ('dbconnect.php');

function getEquipmentUsage() {
  $conn = getDBConnection();

  if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
  }

  $sql = "SELECT Equipment, COUNT(*) as Count FROM Category GROUP BY Equipment";
  $result = $conn->query($sql);

  $equipmentLabels = [
    'B' => 'Recurve/Compound Barebow',
    'C' => 'Compound',
    'L' => 'Longbow',
    'R' => 'Recurve'
  ];

  $data = [];
  if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
      $equipment = $row['Equipment'];
      $equipmentNames = [];
      for ($i = 0; $i < strlen($equipment); $i++) {
        $code = $equipment[$i];
        if (isset($equipmentLabels[$code])) {
          $equipmentNames[] = $equipmentLabels[$code];
        } else {
          // Fetch the equipment description from the EquipmentDescription table
          $equipmentDesc = getEquipmentDescription($conn, $code);
          $equipmentNames[] = $equipmentDesc ? $equipmentDesc : $code;
        }
      }
```

MAJOR – A.I/DATASCIENCE

- PHP function 'getEquipmentUsage()' executes a SQL query to group and count each unique equipment type in the 'Category' table.
- Database connection is established at the start of the function, executing the SQL query that retrieves equipment type and usage count.
- Predefined equipment labels ('B', 'C', 'L', 'R') are used to map codes to full descriptions.
- 'getEquipmentDescription()' function is invoked if an undefined equipment code is encountered, fetching the equipment description from the database.
- If an equipment description is found, it is used; if not, the code itself is used.
- Post data processing, results are converted to JSON format for compatibility and data visualization/analysis.
- Knowledge of equipment usage assists in inventory management, training program decisions, and identifying equipment preference trends.







GRAPHS VISUALISATION

MAJOR – SOFTWARE DEVELOPMENT

This script is for adding, updating, displaying, and deleting archer information from our archery club database.

- addArcherInfo() is a function to add new archers' data to the ArcherInfo table. It accepts the archer's first name, last name, gender, and date of birth (DOB) as arguments. After inserting the data into the database, it will display a success message if the operation is successful or an error message if it fails.
- displayArcherInfo() is a function that retrieves and displays existing archers' data from the database in a tabular form.
- displayArcherSelection() function fetches all the archers from the database and shows them in a dropdown list, providing an option to edit an archer's information.
- displayEditForm() is a function that displays the editable form for an archer's information, pre-filled with the current information of the selected archer. This function also provides the option to delete the archer's record.
- updateArcherInfo() is a function to update an archer's information in the database based on the submitted form data. After updating the record, it will display a success message if the operation is successful or an error message if it fails.
- deleteArcherInfo() is a function that deletes a record from the ArcherInfo table based on the selected archer. After deleting the record, it will display a success message if the operation is successful or an error message if it fails.
- At the end of the script, it checks if the 'edit' button is clicked to display the edit form for the selected archer; otherwise, it shows the archer selection dropdown. Then it calls displayArcherInfo() to display all archers' information.

```php
function displayEditForm($ArcherID) {
    $conn = getDBConnection();

    $sql = "SELECT * FROM ArcherInfo WHERE ArcherID=?";
    $stmt = $conn->prepare($sql);

    if ($stmt === false) {
        die("Error preparing statement: " . $conn->error);
    }

    $stmt->bind_param("i", $ArcherID);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        echo "<form method='post' action=''>";
        echo "First name: <input type='text' name='firstName' value='".$row["FirstName"]."'><br>";
        echo "Last name: <input type='text' name='lastName' value='".$row["LastName"]."'><br>";
        echo "Gender: <input type='text' name='gender' value='".$row["Gender"]."'><br>";
        echo "DOB: <input type='date' name='dob' value='".$row["DOB"]."'><br>";
        echo "<input type='hidden' name='ArcherID' value='".$ArcherID."'>";
        echo "<input type='submit' name='update' value='Update'>";
        echo "<br>";
        echo "<input type='submit' name='delete' value='Delete'>";
        echo "</form>";
    } else {
        echo "No record found";
    }

    $conn->close();
}
```

MAJOR – SOFTWARE DEVELOPMENT

This script includes functions to add, update, delete, display, and select rounds of archery from a database. It uses the POST method to handle the form data. Here's an overview of what each function does:

- addRound(): This function adds a new round to the Rounds table in your database. It uses a prepared statement to prevent SQL injection. The roundName and the number of arrows for each distance (90m, 70m, etc.) are provided as parameters.
- displayRounds(): This function displays all the rounds present in the Rounds table in tabular form.
- getRounds(): This function provides a dropdown list to select a round from all the available rounds in the Rounds table. The selected round's information can then be used for updating or deleting.
- updateRound(): This function updates the number of arrows for each distance for a given round. The roundName is provided as a parameter. Like the addRound function, it uses a prepared statement to prevent SQL injection.
- deleteRound(): This function deletes a specific round from the Rounds table based on the provided roundName.

The main script validates if a form was submitted via POST. Based on the received input, it calls the appropriate function. The script concludes by calling getRounds() and displayRounds() to show the current Rounds table state.

```php
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['updateRound'])) {
    $roundName = isset($_POST['roundName']) ? $_POST['roundName'] : '';
    $arrows90m = isset($_POST['arrows90m']) ? $_POST['arrows90m'] : '';
    $arrows70m = isset($_POST['arrows70m']) ? $_POST['arrows70m'] : '';
    $arrows60m = isset($_POST['arrows60m']) ? $_POST['arrows60m'] : '';
    $arrows50m = isset($_POST['arrows50m']) ? $_POST['arrows50m'] : '';
    $arrows40m = isset($_POST['arrows40m']) ? $_POST['arrows40m'] : '';
    $arrows30m = isset($_POST['arrows30m']) ? $_POST['arrows30m'] : '';
    $arrows20m = isset($_POST['arrows20m']) ? $_POST['arrows20m'] : '';
    $arrows10m = isset($_POST['arrows10m']) ? $_POST['arrows10m'] : '';

    if (!empty($roundName)) { // Check if round name is not empty
        updateRound($roundName, $arrows90m, $arrows70m, $arrows60m, $arrows50m, $arrows40m, $arrows30m, $arrows20m, $arrows10m);
    }
}

// Call getRounds and displayRounds
getRounds();
displayRounds();
```

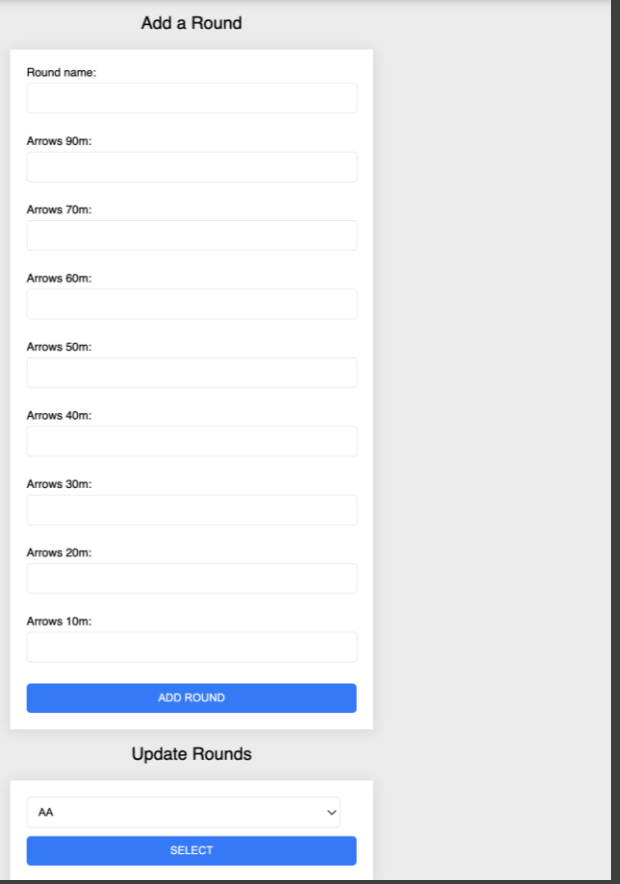
```php
function getClass() {
        $conn = getDBConnection();

        $sql = "SELECT DISTINCT Category
        FROM DefaultEquipment";

        $result = $conn->query($sql);

        if ($result->num_rows > 0) {

            echo "<select name='Class'>";
            while($row = $result->fetch_assoc()) {
                echo "<option value='".$row['Category']."'>".$row['Category']."</option>";
            }
            echo "</select>";
    }
    $conn->close();
}
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['Class'])) {
    $selectedClass = $_POST['Class'];
}
```

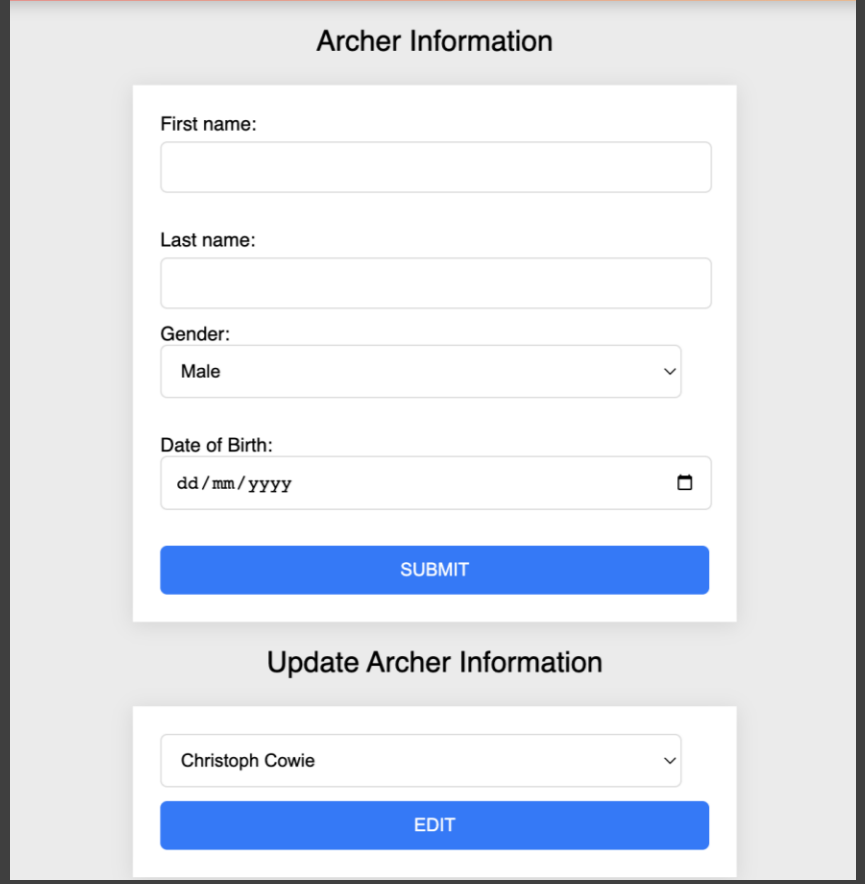
MAJOR – SOFTWARE DEVELOPMENT

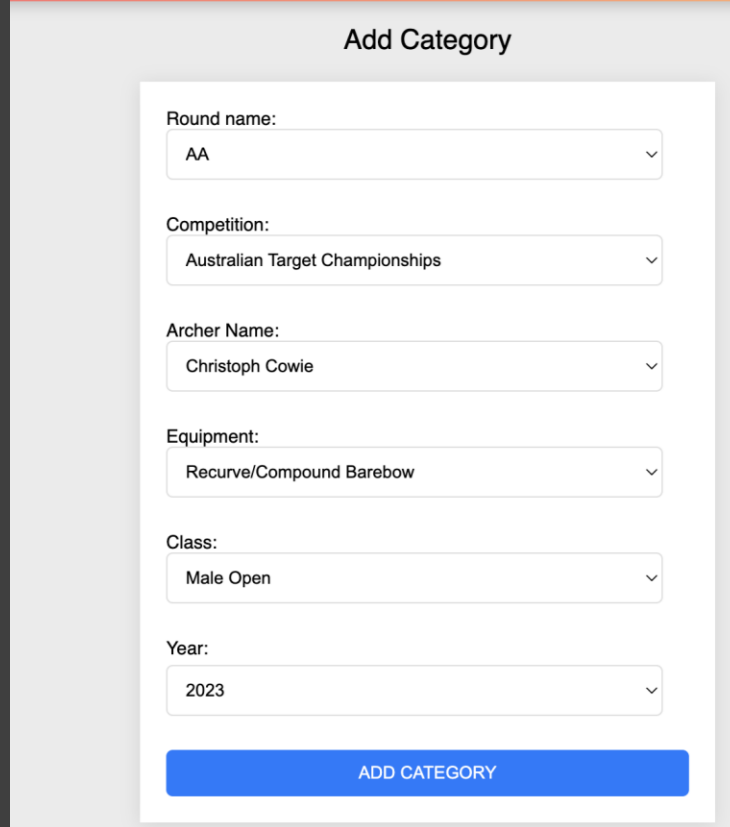
We have defined several functions to fetch competition data, archer names, equipment details, and class categories from the database. Each function fetches relevant data from the database and generates a dropdown menu so that a user can select an option when creating a new category.

- The getCompetition() function retrieves all competitions from the ClubCompetition table and generates a dropdown menu.
- The getArcherName() function retrieves all archer names from the ArcherInfo table and generates another dropdown menu.
- The getEquipment() function retrieves equipment data from the EquipmentDescription table and creates a dropdown menu with it.
- The getClass() function fetches the distinct categories from the DefaultEquipment table and generates a dropdown menu.
- The getYears() function generates options for years from the current year back to 1980.

Once a user submits the form with their selections, the script adds the new category, including the archer's ID, competition ID, round name, class, equipment, and registration year, into the database. On successful addition, a success message appears; in case of failure, the SQL error is displayed.







WEBSITE FEATURES

Implementation & Benefits

- Successfully implemented Club Database for Archery Score Recording
- Centralized platform to monitor their progress and analyze performance
- Promotes competition in a fair environment
- Club recorders are now relieved from the hassle of manual score keeping
- More time can now be dedicated to organizing events and enhancing the archery experience
- The solution fosters transparency and accuracy in scoring and records

SUMMARY

- Our Club Database for Archery Score Recording project signifies a substantial advancement in archery score management.
- The project streamlines the recording process for enhanced efficiency.
- Provides accurate and real-time data to archers and administrators.
- Offers personalized features to improve the user experience.
- Enhances the overall archery experience for both participants and club administrators.
- Envisions a positive impact on our club through this project.
- Aims to inspire others to adopt technological solutions for sports management.