



PROJECT 1: JEOPARDY

BRIAN WALKER



IDENTIFYING THE PROBLEM

- THERE'S NO EASY, WEB-BASED WAY FOR USERS TO PLAY A CUSTOMIZABLE, MULTI-PLAYER JEOPARDY-STYLE TRIVIA GAME WITH AUTOMATED SCORING AND CATEGORY ROTATION.





OBJECTIVES

- IMPLEMENT A TURN-BASED QUESTION-ANSWERING SYSTEM USING PHP FORMS.
- CREATE A PRIZE OR SCORE TRACKING MECHANISM WITH PHP SESSIONS/COOKIES.
- ADD A FEATURE TO RANDOMIZED QUESTIONS WITH DIFFERENT DIFFICULTY LEVELS.
- ENSURE A VISUALLY ENGAGING UI WITH CSS TRANSITIONS/ANIMATIONS



KEY CHALLENGES

- Not allowed to use JavaScript
- Designing system to display questions linked to answers with proper logic
- Implementing randomization of questions with question/answer logic system
- Implementing cookies/sessions to track scores and switch between player 1 and 2
- Marking previously selected questions as used/inaccessible

SOLUTION OVERVIEW

- A PHP-based, browser-friendly Jeopardy game for two players.
- Questions are loaded dynamically from a .txt file for easy editing and scalability.
- Automatically selects 5 random categories per game to keep each session fresh.
- Score is tracked and displayed in real-time, with correct/incorrect answer feedback.
- A live leaderboard is generated as an HTML file for persistent post-game viewing (currently not working via codd server)

SEE IT IN ACTION

JEOPARDY - THE GAME

DEMO RECAP

[HTTPS://GITHUB.COM/IAMBRIANWALKER/JEOPARDY.GIT](https://github.com/IAMBRIANWALKER/JEOPARDY.GIT)

CODE ARCHITECTURE

- Built using PHP with HTML for rendering and basic CSS for styling.
- State is tracked using PHP sessions (`$_SESSION`) to manage game progress.
- Game logic includes:
 - Reading and parsing a question file.
 - Generating a random game board.
 - Handling player turns and score updates.
 - Exporting a live leaderboard to an HTML file

LOAD QUESTIONS FROM .TXT FILE

Key Function: loadQuestionsFromFile()

Content:

- Reads questions.txt where each line =
Category | Value | Question | Answer
- Parses, trims, and stores into an array of question objects.
- Extracts all unique categories and returns both.

```
// Load questions and extract unique categories
1 reference
function loadQuestionsFromFile($filename): array {
    $pool = [];
    $allCategories = [];

    if (!file_exists(filename: $filename)) return [[], []];

    $lines = file(filename: $filename, flags: FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
    foreach ($lines as $line) {
        list($category, $value, $question, $answer) = explode(separator: '|', string: $line);
        $category = trim(string: $category);
        $value = intval(value: $value);
        $question = trim(string: $question);
        $answer = trim(string: $answer);

        $pool[] = [
            'category' => $category,
            'value' => $value,
            'question' => $question,
            'answer' => $answer
        ];

        $allCategories[$category] = true;
    }

    shuffle(array: &$amp;pool);
    return [$pool, array_keys(array: $allCategories)];
}
```

SESSION BASED GAME SETUP

- First-time load: initializes session variables:
 - players, currentPlayer, usedQuestions, questions, answers
- Picks 5 random categories from all available ones.
- Populates \$board and \$answers arrays from the full pool.

```
// Randomly select 5 categories on new game
if (!isset($_SESSION['selectedCategories'])) {
    shuffle(array: &$allCategories);
    $_SESSION['selectedCategories'] = array_slice(array: $allCategories, offset: 0, length: 5);
}
$selectedCategories = $_SESSION['selectedCategories'];
$values = [100, 200, 300, 400, 500];
```

HANDLING ANSWERS AND SWITCHING TURNS

When a user submits an answer:

- It checks correctness (case-insensitive).
- Updates the player's score.
- Saves question to used list.
- Exports scores to leaderboard.html.
- Switches turn.

```
// Handle answer submission
if (isset($_POST['submit_answer'])) {
    $category = $_POST['category'];
    $value = $_POST['value'];
    $answer = trim(string: $_POST['answer']);
    $correct = $_SESSION['answers'][$category][$value] ?? '';

    if (strcasecmp(string1: $answer, string2: $correct) == 0) {
        $_SESSION['players'][$_SESSION['currentPlayer']] += $value;
        $resultMessage = "Correct! +$value points.";
    } else {
        $_SESSION['players'][$_SESSION['currentPlayer']] -= $value;
        $resultMessage = "Incorrect! -$value points. Correct answer: $correct";
    }

    $_SESSION['usedQuestions'][] = "$category-$value";

    // Export leaderboard
    $html = "<?DOCTYPE html>\n<html>\n<head><title>Jeopardy Leaderboard</title></head>\n<body>\n<h1>Jeopardy Leaderboard</h1>\n<ul>\n";
    foreach ($_SESSION['players'] as $player => $score) {
        $html .= "<li><strong>" . htmlspecialchars(string: $player) . ":</strong> " . intval(value: $score) . " points</li>\n";
    }
    $html .= "</ul>\n</body>\n</html>";
    file_put_contents(filename: "leaderboard.html", data: $html);

    // Switch player
    $_SESSION['currentPlayer'] = $_SESSION['currentPlayer'] == 'Player 1' ? 'Player 2' : 'Player 1';
}
```

GAME INTERFACE LOGIC

- Dynamic HTML generation based on current game state.
- Questions are shown only when a value is clicked.
- Used questions are grayed out / disabled.
- Game board is rebuilt each page load using session state.

```
<?php if ($showQuestion): ?>
    <div class="question-box">
        <h2><?= htmlspecialchars(string: $selectedCategory) ?> - <?= $selectedValue ?></h2>
        <p><?= $questionText ?></p>
        <form method="post">
            <input type="hidden" name="category" value="<?= htmlspecialchars(string: $selectedCategory) ?>">
            <input type="hidden" name="value" value="<?= htmlspecialchars(string: $selectedValue) ?>">
            <label>Your Answer:</label>
            <input type="text" name="answer" required>
            <br><br>
            <button type="submit" name="submit_answer">Submit Answer</button>
        </form>
    </div>
<?php else: ?>
    <div class="board">
        <?php foreach ($selectedCategories as $cat): ?>
            <div class="cell category"><?= htmlspecialchars(string: $cat) ?></div>
        <?php endforeach; ?>

        <?php foreach ($values as $val): ?>
            <?php foreach ($selectedCategories as $cat): ?>
                <?php $key = "$cat-$val"; ?>
                <div class="cell">
                    <?php if (!in_array(needle: $key, haystack: $_SESSION['usedQuestions'])): ?>
                        <form method="post">
                            <input type="hidden" name="category" value="<?= htmlspecialchars(string: $cat) ?>">
                            <input type="hidden" name="value" value="<?= htmlspecialchars(string: $val) ?>">
                            <button type="submit"><?= $val ?></button>
                        </form>
                    <?php else: ?>
                        <button disabled>Used</button>
                    <?php endif; ?>
                </div>
            <?php endforeach; ?>
        <?php endforeach; ?>
    </div>
</div>
```

GENERATING THE LEADERBOARD

- Every turn triggers an export of scores to leaderboard.html
- HTML is dynamically generated in PHP.
- Allows persistent, viewable results post-game.

```
// Export leaderboard
$html = "<!DOCTYPE html>\n<html>\n<head>\n<title>Leaderboard</title>\n</head>\n<body>\n<h1>Jeopardy Leaderboard</h1>\n<ul>\n";
foreach ($_SESSION['players'] as $player => $score) {
    |   $html .= "<li>\n<strong>" . htmlspecialchars(string: $player) . " :</strong> " . intval(value: $score) . " points</li>\n";
    }
$html .= "</ul>\n</body>\n</html>";
file_put_contents(filename: "leaderboard.html", data: $html);
```

SOURCES

- https://www.w3schools.com/php/php_examples.asp | PHP general manual
- https://www.w3schools.com/php/php_forms.asp | More info on php forms
- <https://www.php.net/manual/en/function.array-slice.php> | Array Slice Function
- <https://www.php.net/manual/en/function.shuffle.php> | Shuffle Function
- <https://www.php.net/manual/en/function.trim.php> | Trim function
- <https://www.php.net/manual/en/function.explode.php> | Explode function
- <https://www.php.net/manual/en/function.file.php> | File function
- https://www.w3schools.com/php/php_sessions.asp | PHP Sessions Info
- https://youtu.be/58_UCvQBz4Y?si=uF8LTWD2ihVm_g5f | PHP Sessions Tutorial
- https://youtu.be/ALaSX1g9H6E?si=mKxZQfrL_TRxfai | HTML CSS PHP Jeopardy Project Example
- ChatGPT | Creating questions and categories because I'm lazy



THANK YOU