Magic 8-Ball Interactive Web Project

Explained for Beginners

January 29, 2025

1 Introduction

This is an interactive website created with **PHP, HTML, CSS, and JavaScript** that simulates a **Magic 8-Ball**.

- The user asks a question, and the **Magic 8-Ball** responds with a random, funny answer.
- The 8-Ball **shakes, glows, talks**, and **remembers past questions**.

2 How Does It Work?

The project is divided into three main parts:

1. **PHP (Backend - The Brain)** – Chooses a random response and stores question history.
2. **HTML (Frontend - The Body)** – Creates the webpage structure.
3. **CSS & JavaScript (Effects)** – Adds animations (shaking, glowing) and voice responses.

3 PHP: The Brain of the 8-Ball

3.1 1 Starting a PHP Session (Memory System)

PHP uses **sessions** to store information even after the page reloads:

Listing 1: Starting a session

```
session_start();
```

3.2 2 Resetting History

If the "Reset" button is pressed, all stored questions are deleted:

Listing 2: Reset history

```
if (isset($_POST['reset'])) {
    $_SESSION['history'] = [];
    header("Location: " . $_SERVER['PHP_SELF']);
    exit;
}
```

3.3 3 List of Funny Responses

PHP randomly picks from a list of possible answers:

Listing 3: Array of responses

```
$responses = [
    "Yes, but only if you bring me coffee      .",
    "Nope, not today. Ask again when I'm in a good mood          .",
    "Absolutely! But you owe me a pizza          .",
    "Maybe... but do you really want to know the answer?          ",
    "Sure, why not?                    ",
    "Ask again after you do 10 push-ups          .",
    "100% yes! The PHP gods have spoken          .",
    "No way! Not in this universe          .",
    "Only if you say 'PHP is the best' three times          .",
    "Yes, but don't tell anyone I told you... it's top secret          ."
];
```

3.4 4 Selecting a Random Answer

Listing 4: Function to get a random response

```
function getMagic8BallResponse() {
    global $responses;
    return $responses[array_rand($responses)];
}
```

3.5 5 Storing Questions and Answers

Listing 5: Storing session history

```
if (!isset($_SESSION['history'])) {
    $_SESSION['history'] = [];
}
```

```
$question = isset($_POST['question']) ? trim($_POST['question']) : "";
$answer = "";

if (!empty($question)) {
    $answer = getMagic8BallResponse();
    $_SESSION['history'][] = ["question" => $question, "answer" => $answer];
}
```

4 HTML: The Webpage Layout

HTML creates the page elements:

- A text input box for the question.
- A submit button.
- The 8-Ball image.
- The response and history.
- A reset button.

4.1 1 The 8-Ball Image

Listing 6: 8-Ball display

```
<div class="magic-ball" id="ball">        </div>
```

4.2 2 The Question Input Box & Button

Listing 7: User input and submit button

```
<form method="POST" onsubmit="shakeBall();">
    <input type="text" name="question" placeholder="Type␣your␣question␣here..."
    <br>
    <button type="submit">Ask the 8-Ball!</button>
</form>
```

4.3 3 Displaying the Answer

Listing 8: Displaying response

```
<?php if (!empty($question)): ?>
    <div class="response">
        <p>        You asked: <strong><?php echo htmlspecialchars($question); ?><
        <p>          Magic 8−Ball says: <strong id=”answer”><?php echo $answer; ?><
    </div>
    <script>
        setTimeout(() => {
            glowBall();
            speakAnswer(”<?php␣echo␣addslashes($answer);␣?>”);
        }, 500);
    </script>
<?php endif; ?>
```

5 CSS: Making It Look Cool

CSS makes the 8-Ball look real and adds animations.

5.1 1 Shake Effect

Listing 9: Shake animation

```
@keyframes shake {
    0% { transform: rotate(0deg); }
    20% { transform: rotate(10deg); }
    40% { transform: rotate(−10deg); }
    60% { transform: rotate(10deg); }
    80% { transform: rotate(−10deg); }
    100% { transform: rotate(0deg); }
}
.shake {
    animation: shake 0.5s ease−in−out;
}
```

6 JavaScript: Adding Interactivity

JavaScript makes the 8-Ball interactive.

6.1 1 Shake Effect

Listing 10: Shaking effect

```
function shakeBall() {
    let ball = document.getElementById(”ball”);
    ball.classList.add(”shake”);
    setTimeout(() => {
        ball.classList.remove(”shake”);
    }, 500);
}
```

6.2 2 Making the Ball Glow

Listing 11: Glowing effect

```
function glowBall() {
    let ball = document.getElementById("ball");
    ball.classList.add("glow");
    setTimeout(() => {
        ball.classList.remove("glow");
    }, 1000);
}
```

6.3 3 Speaking the Answer

Listing 12: Speech synthesis

```
function speakAnswer(text) {
    let speech = new SpeechSynthesisUtterance();
    speech.text = text;
    window.speechSynthesis.speak(speech);
}
```

7 Conclusion

- **PHP** chooses answers and stores history.
- **HTML** structures the webpage.
- **CSS** makes the 8-Ball look realistic (shaking, glowing).
- **JavaScript** adds interactivity (shaking, glowing, speaking).

Now you understand how the Magic 8-Ball works!