

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Janamdin Mubarak Ho, Tabarak!</title>
  <!-- Tailwind CSS CDN for styling -->
  <script src="https://cdn.tailwindcss.com"></script>
  <!-- PWA Manifest Link -->
  <link rel="manifest" href="/manifest.json">
  <style>
    /* Custom styles for animations and overall look */
    @import
url('https://fonts.googleapis.com/css2?family=Inter:wght@400;700&family=Dancing+Script:wght@700&family=Roboto:wght@400;700&family=Open+Sans:wght@400;700&family=Lato:wght@400;700&family=Montserrat:wght@400;700&family=Merriweather:wght@400;700&family=Playfair+Display:wght@400;700&display=swap');

    body {
      font-family: 'Inter', sans-serif;
      overflow: hidden;
      /* Hide scrollbars due to confetti */
      background: linear-gradient(135deg, #fceabb, #f8b500);
      /* Golden gradient background */
    }

    #app-container {
      min-height: 100vh;
      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      position: relative;
      z-index: 1;
      /* Ensure content is above canvas */
      padding: 1rem;
      /* Added padding for smaller screens */
    }

    #three-container {
      position: absolute;
      top: 0;
      left: 0;
      width: 100%;
      height: 100%;

```

```

    z-index: 0;
    /* Ensure canvas is behind content */
    overflow: hidden;
}

.message-box {
    background-color: rgba(255, 255, 255, 0.98);
    padding: 2rem;
    border-radius: 2rem;
    box-shadow: 0 15px 35px rgba(0, 0, 0, 0.3),
        /* Deeper shadow */
        0 0 0 5px rgba(255, 255, 255, 0.5) inset;
    /* Inner light border */
    text-align: center;
    max-width: 90%;
    /* Max width on mobile */
    width: 100%;
    /* Ensure it takes full available width */
    z-index: 10;
    position: relative;
    transform: scale(0.9);
    opacity: 0;
    animation: fadeInScale 1s forwards cubic-bezier(0.2, 0.8, 0.2,
1);
    animation-delay: 0.5s;
    box-sizing: border-box;
    /* Include padding in element's total width and height */
}

.confetti-container {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    overflow: hidden;
    pointer-events: none;
    /* Allow clicks to pass through */
    z-index: 5;
}

.confetti {
    position: absolute;
    width: 10px;
    height: 10px;
    background-color: var(--color);
    border-radius: 50%;
    opacity: 0;

```

```

    animation: confetti-fall 3s ease-out forwards;
    animation-delay: var(--delay);
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

@keyframes confetti-fall {
  0% {
    transform: translateY(-100vh) rotate(0deg);
    opacity: 0;
  }

  10% {
    opacity: 1;
  }

  100% {
    transform: translateY(100vh) rotate(720deg);
    opacity: 0;
  }
}

@keyframes fadeInScale {
  to {
    opacity: 1;
    transform: scale(1);
  }
}

.animated-text span {
  display: inline-block;
  opacity: 0;
  transform: translateY(20px);
  animation: letter-appear 0.6s forwards cubic-bezier(0.2, 0.8,
0.2, 1);
}

.animated-text span:nth-child(even) {
  animation-delay: 0.1s;
}

.animated-text span:nth-child(odd) {
  animation-delay: 0.2s;
}

.animated-text span:nth-child(3n) {
  animation-delay: 0.3s;
}

```

```

@keyframes letter-appear {
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

#surprise-message {
  max-height: 0;
  overflow: hidden;
  transition: max-height 0.8s ease-in-out, opacity 0.8s
ease-in-out;
  opacity: 0;
  overflow-y: auto;
  /* Enable vertical scroll if content overflows */
  max-width: 100%;
  /* Ensure it doesn't overflow horizontally */
  box-sizing: border-box;
  /* Include padding in element's total width and height */
  text-align: left;
  /* Align text to left for better readability in long messages */
  --glow-color: #ff0;
  /* Default glow color */
}

#surprise-message.show {
  max-height: 500px;
  /* Increased max-height to accommodate more text */
  opacity: 1;
}

/* Glow for the message box */
.glow-effect-box {
  box-shadow: 0 0 15px var(--glow-color);
  transition: box-shadow 0.3s ease-in-out;
}

.rainbow-glow-box {
  animation: rainbow-glow-animation 4s infinite alternate;
}

/* Glow for individual letters */
.text-glow-effect {
  text-shadow: 0 0 8px var(--text-glow-color, #fff),
    /* Base glow */
    0 0 15px var(--text-glow-color, #fff);
  /* Stronger glow */
  transition: text-shadow 0.3s ease-in-out;
}

```

```

}

.text-rainbow-glow {
  animation: text-rainbow-glow-animation 4s infinite alternate;
}

@keyframes rainbow-glow-animation {
  0% {
    box-shadow: 0 0 20px 5px rgba(255, 0, 0, 0.7);
  }

  16% {
    box-shadow: 0 0 20px 5px rgba(255, 255, 0, 0.7);
  }

  33% {
    box-shadow: 0 0 20px 5px rgba(0, 255, 0, 0.7);
  }

  50% {
    box-shadow: 0 0 20px 5px rgba(0, 255, 255, 0.7);
  }

  66% {
    box-shadow: 0 0 20px 5px rgba(0, 0, 255, 0.7);
  }

  83% {
    box-shadow: 0 0 20px 5px rgba(255, 0, 255, 0.7);
  }

  100% {
    box-shadow: 0 0 20px 5px rgba(255, 0, 0, 0.7);
  }
}

@keyframes text-rainbow-glow-animation {
  0% {
    text-shadow: 0 0 10px rgba(255, 0, 0, 0.8);
  }

  16% {
    text-shadow: 0 0 10px rgba(255, 255, 0, 0.8);
  }

  33% {
    text-shadow: 0 0 10px rgba(0, 255, 0, 0.8);
  }
}

```

```

50% {
  text-shadow: 0 0 10px rgba(0, 255, 255, 0.8);
}

66% {
  text-shadow: 0 0 10px rgba(0, 0, 255, 0.8);
}

83% {
  text-shadow: 0 0 10px rgba(255, 0, 255, 0.8);
}

100% {
  text-shadow: 0 0 10px rgba(255, 0, 0, 0.8);
}
}

/* Responsive adjustments for smaller screens */
@media (max-width: 640px) {
  .message-box {
    padding: 1.5rem;
    /* Slightly less padding on small screens */
  }

  .text-5xl {
    font-size: 3.5rem;
    /* Adjust font size for smaller screens */
  }

  .text-2xl {
    font-size: 1.75rem;
    /* Adjust font size for smaller screens */
  }

  .px-8 {
    padding-left: 1.5rem;
    padding-right: 1.5rem;
  }

  .py-3 {
    padding-top: 0.75rem;
    padding-bottom: 0.75rem;
  }
}
</style>
</head>

```

```

<body class="bg-yellow-300">
  <div id="app-container" class="relative overflow-hidden">
    <!-- Three.js Canvas Container -->
    <div id="three-container"></div>

    <!-- Confetti Container -->
    <div id="confetti-container" class="confetti-container"></div>

    <!-- Main Content Box -->
    <div class="message-box space-y-6">
      <!-- Toggle Controls Button -->
      <div class="flex justify-center mb-4">
        <button id="toggle-controls-button" class="px-6 py-2
bg-blue-500 text-white font-bold rounded-full shadow-md
hover:bg-blue-600 transition-all duration-300 ease-in-out
focus:outline-none focus:ring-4 focus:ring-blue-300">
          ⚙️ Controls
        </button>
      </div>

      <!-- All Controls Container (initially hidden) -->
      <div id="all-controls-container" class="flex flex-col
sm:flex-row items-center justify-center gap-4 mb-4" style="display:
none;">
        <!-- Language and Font Selectors -->
        <select id="language-select" class="p-2 rounded-md border
border-gray-300 shadow-sm focus:ring-indigo-500
focus:border-indigo-500">
          <option value="hinglish">Hin-English</option>
          <option value="english">English</option>
          <option value="marathi">Marathi</option>
        </select>
        <select id="font-select" class="p-2 rounded-md border
border-gray-300 shadow-sm focus:ring-indigo-500
focus:border-indigo-500">
          <option value="Dancing Script">Dancing Script</option>
          <option value="Inter">Inter</option>
          <option value="Roboto">Roboto</option>
          <option value="Open Sans">Open Sans</option>
          <option value="Lato">Lato</option>
          <option value="Montserrat">Montserrat</option>
          <option value="Merriweather">Merriweather</option>
          <option value="Playfair Display">Playfair Display</option>
        </select>

        <!-- Read Aloud and Glow Options -->
        <button id="read-aloud-button" class="px-6 py-2 bg-green-500

```

```

text-white font-bold rounded-full shadow-md hover:bg-green-600
transition-all duration-300 ease-in-out focus:outline-none
focus:ring-4 focus:ring-green-300">
    <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6
inline-block mr-2" fill="none" viewBox="0 0 24 24"
stroke="currentColor">
    <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15.536 8.464a5 5 0 010
7.072m2.828-9.911.414-1.414M7.071 17.657l-1.414 1.414M4.929
6.364L3.515 4.95M12 20.25v-1.5m0-15v-1.5m-7.778 12.129l1.061
1.061M1.75 12h1.5m14.5 0h1.5M4.293 4.293l1.061 1.061M18.939
18.939l1.061 1.061M8.04 15.334a4 4 0 00-1.85-3.082L3
8.354V6.5h3.04l3.197 2.656a4 4 0 001.076 6.377z" />
    </svg>
    पढ़कर सुनाओ
</button>

<button id="toggle-glow-button" class="px-6 py-2 bg-yellow-500
text-white font-bold rounded-full shadow-md hover:bg-yellow-600
transition-all duration-300 ease-in-out focus:outline-none
focus:ring-4 focus:ring-yellow-300">
    ✨ Toggle Glow
</button>
<input type="color" id="glow-color-picker" value="#FFFF00"
class="w-10 h-10 p-1 rounded-full border-none cursor-pointer">
    <button id="multi-color-glow-button" class="px-6 py-2
bg-red-500 text-white font-bold rounded-full shadow-md
hover:bg-red-600 transition-all duration-300 ease-in-out
focus:outline-none focus:ring-4 focus:ring-red-300">
    🌈 Multi-Color Glow
</button>
</div>

<h1 id="main-title" class="text-5xl md:text-6xl font-extrabold
text-transparent bg-clip-text bg-gradient-to-r from-pink-500
to-red-600 mb-4 font-['Dancing_Script']">
    Happy Birthday! 🎂
</h1>
<p id="main-greeting" class="text-2xl md:text-3xl text-gray-800
animated-text">
    <span>Tabarak</span> <span>Khan,</span> <span>Bahut</span>
<span>Bahut</span> <span class="font-bold
text-pink-600">Janamdin</span> <span>Mubarak!</span> 🎉
</p>

<button id="surprise-button" class="mt-8 px-8 py-3
bg-gradient-to-r from-purple-600 to-indigo-700 text-white font-bold

```



```

text-xl rounded-full shadow-lg hover:shadow-xl transform
hover:scale-105 transition-all duration-300 ease-in-out cursor:pointer
focus:outline-none focus:ring-4 focus:ring-purple-300">
    Ek Khaas Tohfe Ke Liye Click Karein! 🎁
</button>

```

```

<div id="surprise-message" class="mt-6 text-xl text-gray-700
leading-relaxed p-4 bg-yellow-100 rounded-lg border
border-yellow-300">

```

```

    <!-- Content will be dynamically inserted here -->
</div>

```

```

<!-- Credit Section -->
<div class="mt-8 text-center text-gray-500 text-sm">
    Created by <span class="font-bold text-blue-600">Mushahid
Raza</span> ❤️
</div>
</div>

```

```

<!-- Three.js CDN -->
<script
src="https://cdn.jsdelivr.net/npm/three.js/r128/three.min.js
"></script>
<script>
    // Ensure the entire page is loaded before running the script
    window.onload = function() {
        const threeContainer =
document.getElementById('three-container');
        const surpriseButton =
document.getElementById('surprise-button');
        const surpriseMessage =
document.getElementById('surprise-message');
        const confettiContainer =
document.getElementById('confetti-container');
        const mainTitle = document.getElementById('main-title');
        const mainGreeting = document.getElementById('main-greeting');
        const languageSelect =
document.getElementById('language-select');
        const fontSelect = document.getElementById('font-select');
        const readAloudButton =
document.getElementById('read-aloud-button');
        const readAloudContainer =
document.getElementById('read-aloud-container');
        const toggleGlowButton =
document.getElementById('toggle-glow-button');
        const glowColorPicker =

```

```

document.getElementById('glow-color-picker');
const multiColorGlowButton =
document.getElementById('multi-color-glow-button');
const toggleControlsButton =
document.getElementById('toggle-controls-button');
const allControlsContainer =
document.getElementById('all-controls-container');

let glowActive = false;
let multiColorGlowActive = false;

// --- Language Content ---
const messages = {
  'hinglish': {
    title: "Happy Birthday! 🎂",
    greeting: "<span>Tabarak</span> <span>Khan,</span>
<span>Bahut</span> <span>Bahut</span> <span class=\"font-bold
text-pink-600\">Janamdin</span> <span>Mubarak!</span> 🎉",
    button: "Ek Khaas Tohfe Ke Liye Click Karein! 📺",
    readAloud: "पढ़कर सुनाओ",
    surprise: `
      <p>Mere Azeez Dost Tabarak Khan, 🙌</p>
      <p class="mt-2">Aapko saalgirah ki dhero'n mubarakbad! 🎊
Allah aapko lambi umar, khoob saari khushiya'n, kamyabi aur sehat ata
farmaye. 💖 Aapki zindagi mein hamesha raunaq bani rahe, aur har din
ek naya jashn ho. ✨</p>
      <p class="mt-4 font-bold text-red-700">Khush rahiye aur
muskurate rahiye! 😊</p>
      <p class="mt-6 text-lg font-semibold text-gray-600">Aapka
sacha dost, <span class="text-blue-600">Mushahid Raza</span> ki taraf
se 🙏</p>
      <p class="mt-2">Yeh message khaas aapke liye likha gaya
hai, taaki aapka yeh din aur bhi yaadgaar ban sake. 🌟 Har mushkil
mein main aapke saath khada rahoonga, aur har khushi mein aapka
humsafar banoonga. Aapki dosti mere liye anmol hai. 💎 Allah aapko har
buri nazar se bachaye aur aapki har jaiz khwahish पूरी kare. 🎯 Duaon
mein yaad rakhiyega. 🙏</p>
      <p class="mt-4 text-sm text-gray-500">Note: Yeh digital
tohfa sirf aapke liye banaya gaya hai. Ummeed hai aapko pasand aayega.
❤️</p>
    `,
  },
  'english': {
    title: "Happy Birthday! 🎂",
    greeting: "<span>Tabarak</span> <span>Khan,</span>
<span>Many</span> <span>Many</span> <span class=\"font-bold
text-pink-600\">Happy</span> <span>Returns</span> <span>of</span>

```

```

<span>the</span> <span>Day!</span> 🎉",
  button: "Click for a Special Gift! 📺",
  readAloud: "Read Aloud",
  surprise: `
    <p>My Dear Friend Tabarak Khan, 🙌</p>
    <p class="mt-2">Many happy returns of the day to you! 🎂
May Allah grant you a long life, immense happiness, success, and good
health. 💖 May your life always be vibrant, and every day be a new
celebration. ✨</p>
    <p class="mt-4 font-bold text-red-700">Stay happy and keep
smiling! 😊</p>
    <p class="mt-6 text-lg font-semibold text-gray-600">From
your true friend, <span class="text-blue-600">Mushahid Raza</span>
🙏</p>

    <p class="mt-2">This message has been specially written
for you, so that your day can become even more memorable. ☀️ I will
stand by you in every difficulty, and be your companion in every joy.
Your friendship is invaluable to me. 💎 May Allah protect you from all
evil eyes and fulfill your every rightful wish. 🕒 Please remember me
in your prayers. 🙏</p>
    <p class="mt-4 text-sm text-gray-500">Note: This digital
gift has been made especially for you. Hope you like it. ❤️</p>
  `,
  'marathi': {
    title: "वाढदिवसाच्या हार्दिक शुभेच्छा! 🎂",
    greeting: "<span>तबारक</span> <span>खान,</span>
<span>तुम्हाला</span> <span>वाढदिवसाच्या</span> <span>खूप</span>
<span>खूप</span> <span class=\"font-bold text-pink-600\">शुभेच्छा!</span>
🎉",
    button: "एका खास भेटवस्तूसाठी क्लिक करा! 📺",
    readAloud: "मोठ्याने वाचा",
    surprise: `
      <p>माझ्या प्रिय मित्रा तबारक खान, 🙌</p>
      <p class="mt-2">तुम्हाला वाढदिवसाच्या खूप खूप शुभेच्छा! 🎂 अल्लाह
तुम्हाला दीर्घायुष्य, खूप आनंद, यश आणि चांगले आरोग्य देवो. 💖 तुमचे जीवन नेहमी आनंदी
राहो आणि प्रत्येक दिवस एक नवीन उत्सव असो. ✨</p>
      <p class="mt-4 font-bold text-red-700">आनंदी रहा आणि हसत
रहा! 😊</p>
      <p class="mt-6 text-lg font-semibold text-gray-600">तुमचा
खरा मित्र, <span class="text-blue-600">मुशाहिद रझा</span> कडून 🙏</p>
      <p class="mt-2">हा संदेश खास तुमच्यासाठी लिहिला आहे, जेणेकरून तुमचा
हा दिवस आणखी अविस्मरणीय होईल. ☀️ प्रत्येक अडचणीत मी तुमच्यासोबत उभा राहीन आणि
प्रत्येक आनंदात तुमचा सोबती होईन. तुमची मैत्री माझ्यासाठी अमूल्य आहे. 💎 अल्लाह तुम्हाला
सर्व वाईट नजरेपासून वाचवो आणि तुमच्या प्रत्येक योग्य इच्छा पूर्ण करो. 🕒 कृपया मला तुमच्या
प्रार्थनांमध्ये लक्षात ठेवा. 🙏</p>
      <p class="mt-4 text-sm text-gray-500">टीप: ही डिजिटल भेट फक्त
तुमच्यासाठी बनवली आहे. आशा आहे तुम्हाला ती आवडेल. ❤️</p>
    `
  }
}

```

```

    }
};

// --- Font Families ---
const fontFamilies = {
  'Dancing Script': "'Dancing Script', cursive",
  'Inter': "'Inter', sans-serif",
  'Roboto': "'Roboto', sans-serif",
  'Open Sans': "'Open Sans', sans-serif",
  'Lato': "'Lato', sans-serif",
  'Montserrat': "'Montserrat', sans-serif",
  'Merriweather': "'Merriweather', serif",
  'Playfair Display': "'Playfair Display', serif"
};

// Function to update content based on selected language
function updateLanguage(lang) {
  mainTitle.innerHTML = messages[lang].title; // Use innerHTML
  for emojis
    mainGreeting.innerHTML = messages[lang].greeting; // Use
    innerHTML because of span tags and emojis
    surpriseButton.innerHTML = messages[lang].button;
    readAloudButton.innerHTML = `<svg
xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 inline-block mr-2"
fill="none" viewBox="0 0 24 24" stroke="currentColor">
    <path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15.536 8.464a5 5 0 010
7.072m2.828-9.911.414-1.414M7.071 17.657l-1.414 1.414M4.929
6.364L3.515 4.95M12 20.25v-1.5m0-15v-1.5m-7.778 12.129l1.061
1.061M1.75 12h1.5m14.5 0h1.5M4.293 4.293l1.061 1.061M18.939
18.939l1.061 1.061M8.04 15.334a4 4 0 00-1.85-3.082L3
8.354V6.5h3.04l3.197 2.656a4 4 0 001.076 6.377z" />
    </svg>${messages[lang].readAloud}`;
    surpriseMessage.innerHTML = messages[lang].surprise;
  }

// Function to update font based on selected font
function updateFont(fontName) {
  document.body.style.fontFamily = fontFamilies[fontName];
  mainTitle.style.fontFamily = fontFamilies[fontName]; // Apply
  to title too
  mainGreeting.style.fontFamily = fontFamilies[fontName]; //
  Apply to greeting too
  surpriseButton.style.fontFamily = fontFamilies[fontName]; //
  Apply to button
  surpriseMessage.style.fontFamily = fontFamilies[fontName]; //
  Apply to message

```

```

        languageSelect.style.fontFamily = fontFamilies[fontName];
        fontSelect.style.fontFamily = fontFamilies[fontName];
        readAloudButton.style.fontFamily = fontFamilies[fontName];
        toggleGlowButton.style.fontFamily = fontFamilies[fontName];
        multiColorGlowButton.style.fontFamily =
fontFamilies[fontName];
        toggleControlsButton.style.fontFamily =
fontFamilies[fontName];
    }

    // Initial load
    updateLanguage(languageSelect.value);
    updateFont(fontSelect.value); // Set default font

    // Event listeners for language and font selection
    languageSelect.addEventListener('change', (event) => {
        updateLanguage(event.target.value);
    });

    fontSelect.addEventListener('change', (event) => {
        updateFont(event.target.value);
    });

    // --- Three.js Setup (3D Animation) ---
    let scene, camera, renderer, textMesh, spheres = [];
    const sphereCount = 50;

    function initThree() {
        // Scene
        scene = new THREE.Scene();
        scene.background = null; // Transparent background
        // Camera
        camera = new THREE.PerspectiveCamera(75, window.innerWidth /
window.innerHeight, 0.1, 1000);
        camera.position.z = 5;
        // Renderer
        renderer = new THREE.WebGLRenderer({
            antialias: true,
            alpha: true
        }); // Alpha true for transparent background
        renderer.setSize(window.innerWidth, window.innerHeight);
        threeContainer.appendChild(renderer.domElement);
        // Lighting
        const ambientLight = new THREE.AmbientLight(0xffffff, 0.5);
        scene.add(ambientLight);
        const directionalLight = new THREE.DirectionalLight(0xffffff,

```

```

0.8);
    directionalLight.position.set(0, 1, 1).normalize();
    scene.add(directionalLight);
    // --- 3D "Birthday" Text ---
    const loader = new THREE.FontLoader();
    // Using a generic font, as external font loading for Three.js
in this environment is complex.
    // For better fonts, a font JSON file would be loaded.
    // Here, a simple "Arial" type representation is used.

loader.load('https://raw.githubusercontent.com/mrdoob/three.js/dev/exa
mples/fonts/helvetiker_regular.typeface.json', function(font) {
    const textGeometry = new THREE.TextGeometry('Birthday', {
        font: font,
        size: 0.8,
        height: 0.2,
        curveSegments: 12,
        bevelEnabled: true,
        bevelThickness: 0.03,
        bevelSize: 0.02,
        bevelOffset: 0,
        bevelSegments: 5
    });
    textGeometry.computeBoundingBox();
    const centerOffset = -0.5 * (textGeometry.boundingBox.max.x
- textGeometry.boundingBox.min.x);
    textGeometry.translate(centerOffset, 0, 0); // Center the
text

    const textMaterial = new THREE.MeshPhongMaterial({
        color: 0xffd700,
        emissive: 0xccaa00,
        specular: 0xffffffff,
        shininess: 100
    });
    textMesh = new THREE.Mesh(textGeometry, textMaterial);
    textMesh.position.y = 1; // Position slightly above center
    scene.add(textMesh);
});
// --- Floating Spheres (Balloons/Bubbles) ---
const sphereGeometry = new THREE.SphereGeometry(0.1, 16, 16);
const colors = [0xff0000, 0x00ff00, 0x0000ff, 0xffff00,
0xff00ff, 0x00ffff, 0xffa500]; // Various bright colors
for (let i = 0; i < sphereCount; i++) {
    const material = new THREE.MeshPhongMaterial({
        color: colors[Math.floor(Math.random() * colors.length)]
    });
    const sphere = new THREE.Mesh(sphereGeometry, material);
    sphere.position.set(

```

```

        (Math.random() - 0.5) * 10,
        (Math.random() - 0.5) * 10,
        (Math.random() - 0.5) * 10
    );
    sphere.velocity = new THREE.Vector3(
        (Math.random() - 0.5) * 0.01,
        (Math.random() - 0.5) * 0.01,
        (Math.random() - 0.5) * 0.01
    );
    scene.add(sphere);
    spheres.push(sphere);
}
// Handle window resizing
window.addEventListener('resize', onWindowResize, false);
}

function onWindowResize() {
    camera.aspect = window.innerWidth / window.innerHeight;
    camera.updateProjectionMatrix();
    renderer.setSize(window.innerWidth, window.innerHeight);
}

function animateThree() {
    requestAnimationFrame(animateThree);
    // Rotate the 3D text
    if (textMesh) {
        textMesh.rotation.y += 0.005;
        textMesh.rotation.x += 0.002;
    }
    // Animate spheres
    spheres.forEach(sphere => {
        sphere.position.add(sphere.velocity);
        // Bounce off boundaries
        if (sphere.position.x > 5 || sphere.position.x < -5)
sphere.velocity.x *= -1;
        if (sphere.position.y > 5 || sphere.position.y < -5)
sphere.velocity.y *= -1;
        if (sphere.position.z > 5 || sphere.position.z < -5)
sphere.velocity.z *= -1;
    });
    renderer.render(scene, camera);
}
initThree();
animateThree();

// --- Confetti Animation ---
function createConfetti() {
    const colors = ['#f0f', '#0ff', '#ff0', '#f00', '#0f0',

```

```

'#00f', '#ffa500'];
    for (let i = 0; i < 150; i++) {
        const confetti = document.createElement('div');
        confetti.classList.add('confetti');
        confetti.style.setProperty('--color',
colors[Math.floor(Math.random() * colors.length)]);
        confetti.style.setProperty('--delay', `${Math.random() *
2}s`);
        confetti.style.left = `${Math.random() * 100}vw`;
        confettiContainer.appendChild(confetti);
        // Remove confetti after animation to prevent DOM bloat
        confetti.addEventListener('animationend', () => {
            confetti.remove();
        });
    }
}

// --- Surprise Button Logic ---
surpriseButton.addEventListener('click', () => {
    // Show the surprise message
    surpriseMessage.classList.add('show');
    surpriseButton.style.display = 'none'; // Hide the button
after click
    createConfetti();
});

// --- Text-to-Speech Logic ---
readAloudButton.addEventListener('click', () => {
    if ('speechSynthesis' in window) {
        const textToSpeak = surpriseMessage.textContent;
        const utterance = new SpeechSynthesisUtterance(textToSpeak);

        // Set language for better pronunciation (optional, adjust
as needed)
        const currentLang = languageSelect.value;
        if (currentLang === 'english') {
            utterance.lang = 'en-US';
        } else if (currentLang === 'marathi') {
            utterance.lang = 'mr-IN'; // Marathi (India)
        } else {
            // For Hinglish, a neutral or Hindi voice might work best,
or English
            utterance.lang = 'hi-IN'; // Hindi (India)
        }

        speechSynthesis.speak(utterance);
    } else {
        console.error('Speech Synthesis not supported in this

```



```

browser.');
```

```

    // Use a custom message box instead of alert()
    const messageDiv = document.createElement('div');
    messageDiv.className = 'fixed top-1/2 left-1/2
-translate-x-1/2 -translate-y-1/2 bg-red-500 text-white p-4 rounded-lg
shadow-lg z-50';
    messageDiv.textContent = 'Sorry, your browser does not
support text-to-speech.';
    document.body.appendChild(messageDiv);
    setTimeout(() => {
        messageDiv.remove();
    }, 3000); // Remove message after 3 seconds
}
});

// --- Glow Logic ---
function applyGlow() {
    // Remove all glow classes first
    surpriseMessage.classList.remove('glow-effect-box',
'rainbow-glow-box');
    Array.from(surpriseMessage.querySelectorAll('p,
span')).forEach(el => {
        el.classList.remove('text-glow-effect',
'text-rainbow-glow');
    });

    if (glowActive) {
        // Apply glow to the box
        // surpriseMessage.classList.add('glow-effect-box'); //
Decide if box should glow
        if (multiColorGlowActive) {
            surpriseMessage.classList.add('rainbow-glow-box'); // Box
rainbow glow
            Array.from(surpriseMessage.querySelectorAll('p,
span')).forEach(el => {
                el.classList.add('text-rainbow-glow'); // Text rainbow
glow
            });
        } else {
            surpriseMessage.classList.add('glow-effect-box'); // Box
single color glow
            surpriseMessage.style.setProperty('--glow-color',
glowColorPicker.value);
            Array.from(surpriseMessage.querySelectorAll('p,
span')).forEach(el => {
                el.classList.add('text-glow-effect'); // Text single
color glow
                el.style.setProperty('--text-glow-color',

```

```

glowColorPicker.value);
    });
  }
}
// If glowActive is false, all classes are already removed by
the initial step.
}

toggleGlowButton.addEventListener('click', () => {
  glowActive = !glowActive;
  // If toggling off, ensure multi-color is also off
  if (!glowActive) {
    multiColorGlowActive = false;
  }
  applyGlow();
});

glowColorPicker.addEventListener('input', () => {
  // If single color glow is active, update the color instantly
  if (glowActive && !multiColorGlowActive) {
    surpriseMessage.style.setProperty('--glow-color',
glowColorPicker.value); // For box glow
    Array.from(surpriseMessage.querySelectorAll('p,
span')).forEach(el => {
      el.style.setProperty('--text-glow-color',
glowColorPicker.value); // For text glow
    });
  }
});

multiColorGlowButton.addEventListener('click', () => {
  multiColorGlowActive = !multiColorGlowActive;
  // If multi-color glow is activated, ensure glow is active
  if (multiColorGlowActive) {
    glowActive = true;
  }
  applyGlow();
});

// --- Controls Toggle Logic ---
toggleControlsButton.addEventListener('click', () => {
  if (allControlsContainer.style.display === 'none' ||
allControlsContainer.style.display === '') {
    allControlsContainer.style.display = 'flex';
  } else {
    allControlsContainer.style.display = 'none';
  }
});

```

```
// Initially hide controls and read aloud/glow options
allControlsContainer.style.display = 'none';

// Show controls only after surprise message is revealed
surpriseButton.addEventListener('click', () => {
  surpriseMessage.classList.add('show');
  surpriseButton.style.display = 'none';
  // You might want to show the toggle controls button here, but
not the controls themselves
  toggleControlsButton.style.display = 'block'; // Or
'inline-block'
  createConfetti();
});

// Ensure initial display state of toggle controls button
toggleControlsButton.style.display = 'none';
};
</script>
</body>

</html>
```