

Assignment 5 Web Technologies
Kuanysh Beibarys SE-2413

Task-1:

Created an array of items with images, rarity levels, and drop chances to dynamically display in the roulette.

```
const items = [
  { src: "/images/videoframe_2042.png", rarity: "shit", chance: 40 },
  { src: "/images/videoframe_2481.png", rarity: "common", chance: 30 },
  { src: "/images/videoframe_2694.png", rarity: "rare", chance: 15 },
  { src: "/images/videoframe_7667.png", rarity: "epic", chance: 10 },
  { src: "/images/videoframe_8916.png", rarity: "legendary", chance: 5 },
];
```

Task-2:

Added sound effects for spinning and for each rarity type, including multiple random sounds per rarity.

```
const soundPaths = {
  spin: "/sounds/startsound.mp3",
  shit: ["/sounds/shit.mp3", "/sounds/shit1.mp3"],
  common: ["/sounds/common.mp3", "/sounds/common.mp3"],
  rare: ["/sounds/rare.mp3", "/sounds/rare1.mp3"],
  epic: ["/sounds/epic.mp3", "/sounds/epic.mp3"],
  legendary: ["/sounds/legendary.mp3", "/sounds/legendary.mp3"],
};
```

Task-3:

Selected and linked all key DOM elements such as carousel, buttons, and popup for interactive control.

```
//Task-3 Selecting DOM Elements
const carousel = document.getElementById("carousel");
const openCase = document.getElementById("openCase");
const winblock = document.getElementById("winblock");
const winImage = document.getElementById("winImage");
const prizeText = document.getElementById("prize");
const itemWidth = 160;
const centerX = 300;
```

Task-4:

Implemented functions to generate random items and fill the roulette dynamically using loops and objects.

```
151 //Task-4 Arrays, Loops, Objects
152 ✓ function getRandomItem() {
153     const totalChance = items.reduce((sum, i) => sum + i.chance, 0);
154     const rand = Math.random() * totalChance;
155     let cumulative = 0;
156     ✓ for (const item of items) {
157         cumulative += item.chance;
158         if (rand <= cumulative) return item;
159     }
160     return items[0];
161 }
162
163 ✓ function fillCarousel() {
164     carousel.innerHTML = "";
165     ✓ for (let i = 0; i < 30; i++) {
166         const randomItem = getRandomItem();
167         const img = document.createElement("img");
168         img.src = randomItem.src;
169         img.dataset.rarity = randomItem.rarity;
170         carousel.appendChild(img);
171     }
172 }
173 | fillCarousel();
174
```

Task-5:

Added a function to play the spinning sound once when the roulette starts.

```
//Task-5 Play start sound (Event handling)
✓ function startSpinSound() {
    const spinSound = sounds.spin;
    spinSound.currentTime = 0;
    spinSound.volume = 0.7;
    spinSound.play().catch(err => console.log("Audio blocked:", err));
}
```

Task-6:

Implemented the main spinning animation using requestAnimationFrame() with gradual deceleration.

```

185 //Task-6 Event Handling + Animation
186 function startSpin() {
187     if (spinning) return;
188     spinning = true;
189     openCase.disabled = true;
190     fillCarousel();
191     startSpinSound();
192
193     let offset = 0;
194     let speed = 40;
195     const spinDuration = 4000 + Math.random() * 2000;
196     const decelerationStart = spinDuration * 0.6;
197     let startTime = performance.now();
198
199     function spin(now) {
200         const elapsed = now - startTime;
201         offset += speed;
202         carousel.style.transform = `translateX(-${offset}px)`;
203         if (offset > carousel.scrollWidth / 2) offset = 0;
204         if (elapsed > decelerationStart) speed *= 0.985;
205         if (elapsed < spinDuration) {
206             requestAnimationFrame(spin);
207         } else {
208             finishSpin(offset);
209         }
210     }
211     requestAnimationFrame(spin);
212 }

```

Task-7:

Calculated the winning item based on the final carousel position and smoothly aligned it to the center.

```

//Task-7 Calculating Winner + Animation Finish
function finishSpin(offset) {
    const imgs = carousel.querySelectorAll("img");
    const totalWidth = imgs.length * itemWidth;
    const centerPos = (offset + centerX) % totalWidth;
    const winnerIndex = Math.floor(centerPos / itemWidth);
    const winnerImg = imgs[winnerIndex];
    const winnerSrc = winnerImg.src;
    const rarity = winnerImg.dataset.rarity;
    const alignOffset = winnerIndex * itemWidth - centerX + itemWidth / 2;
    carousel.style.transition = "transform 0.6s ease-out";
    carousel.style.transform = `translateX(-${alignOffset}px)`;
    setTimeout(() => showWin(winnerSrc, rarity), 600);
}

```

Task-8:

Used a switch-case statement to show different messages and sounds depending on item rarity.

```
229 //Task-8 Switch-case + Callback + Popup Display
230 function showWin(src, rarity) {
231     spinning = false;
232     openCase.disabled = false;
233
234     let rarityMessage;
235     switch (rarity) {
236         case "shit":
237             rarityMessage = "Govno";
238             break;
239         case "common":
240             rarityMessage = "common";
241             break;
242         case "rare":
243             rarityMessage = "Rare";
244             break;
245         case "epic":
246             rarityMessage = "Epic";
247             break;
248         case "legendary":
249             rarityMessage = "LEGENDARY";
250             break;
251         default:
252             rarityMessage = "Unknown rarity!";
253     }
254
255     setTimeout(() => {
256         const possibleSounds = sounds[rarity];
257         if (possibleSounds && possibleSounds.length > 0) {
258             const sound = possibleSounds[Math.floor(Math.random() * possibleSounds.length)];
259             sound.currentTime = 0;
260             sound.volume = 0.7;
261             sound.play().catch(err => console.log("Audio blocked:", err));
262         }
263     }, 100);
```

Task-9:

Added keyboard and mouse event listeners to open the case with Enter or click, and close the popup.

```
270 //Task-9 Additional Event Handling
271 winblock.addEventListener("click", e => {
272     if (e.target === winblock) winblock.style.display = "none";
273 });
274 document.addEventListener("keydown", e => {
275     if (e.key === "Enter") startSpin();
276 });
277 openCase.addEventListener("click", startSpin);
278
```

