



Datei

Geteilte Dateien können nicht angezeigt werden.

Ich brauche ein schlichtes, modernes, ansprechendes Design das zu den hochgeladen Quellcodes passt. Es soll ein dark theme haben und cool aussehen, beinhaltet gerne kleine icons. Es soll so aussehen als wäre es von einem professionellem Designer gemacht worden, es soll also qualitativ gut aussehen, der code soll also keine Fehler haben, lass dir also Zeit beim generieren der passenden Lösung. Übertreib hierbei aber bitte nicht. Beachte bitte, dass im JS ein HTML Code generiert wird, dieser sollte auch entsprechend designed werden. Einmal ist das unter der Funktion pop(), andererseits unter addNewCutBtn.addEventListener. Außerdem sollst du die Klassennamen im HTML beachten, diese sollen nicht verändert werden, da sie im js wieder aufgegriffen werden. Benutze bitte ausschließlich TailwindCSS kein anderes Framework! Gebe mir den neuen HTML Code aus, sowie genau was im JS angepasst werden soll.

⌚ 35 Sekunden nachgedacht >

Hier ist das überarbeitete, moderne Dark-Theme-Design mit TailwindCSS:

index.html (Aktualisiert)

html

```
<!DOCTYPE html>
<html lang="de" class="dark">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>BeLL Videoschnitt-KI</title>
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/heroicons@2.0.12/outline"></script>
  <style>
    @import url('https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap');
  body {
    font-family: 'Inter', sans-serif;
  }
}
```

```
.cutSection {  
    transition: all 0.3s ease;  
}  
.cutSection:hover {  
    transform: translateY(-2px);  
    box-shadow: 0 10px 25px -5px rgba(0, 0, 0, 0.25);  
}  
input[type="range"]::-webkit-slider-thumb {  
    -webkit-appearance: none;  
    appearance: none;  
    width: 16px;  
    height: 16px;  
    border-radius: 50%;  
    background: #3b82f6;  
    cursor: pointer;  
}  
</style>  
</head>  
    <div class="max-w-4xl w-full space-y-8">  
        <header class="text-center mb-12">  
            <h1 class="text-3xl md:text-4xl font-bold bg-gradient-to-r from-blue-400 to-purple-500 bg-clip-text text-transparent">  
                BeLL Videoschnitt-KI  
            </h1>  
            <p class="text-gray-400 mt-2">Automatisches Entfernen von Versprechern und Pausen</p>  
        </header>  
  
        <div class="bg-gray-800 rounded-xl p-6 shadow-lg">  
            <div class="Informationen mb-10">  
                <p class="mb-4">  
                    Helfen Sie mir bei meiner Besonderen Lernleistung!  
                </p>  
                <p class="mb-4">  
                    Mein Ziel ist es, eine KI anzutrainieren, die automatisch Versprecher und Pausen in Videos entfernt.  
                </p>  
                <p class="font-medium mb-2">So funktioniert es:</p>  
                <ol class="list-decimal list-inside ml-4 space-y-2 text-gray-300">  
                    <li>Wählen Sie ein MP4-Video aus.</li>  
                    <li>  
                        Klicken Sie auf „Schnitt hinzufügen“, um die gewünschten Schnittpunkte zu setzen.  
                        <span class="text-gray-400 block">(Zeitformat: MM:SS.ss - Minute:Sekunde. Millisekunde)</span>  
                    </li>  
                </ol>  
            </div>  
        </div>  
    </div>
```

```
</li>
<li>Beschreiben Sie kurz, was Sie geschnitten haben.</li>
<li>Senden Sie das bearbeitete Video ab.</li>
</ol>


Vielen Dank für Ihre Unterstützung!</p>
</div>

<div id="Video" class="flex flex-col items-center mb-8">
  <input type="file" id="VideoInput" accept="video/mp4" class="hidden"/>
  <label for="VideoInput" id="VideoInputLabel"
    class="cursor-pointer bg-blue-600 hover:bg-blue-700 text-white font-me
dium py-3 px-6 rounded-lg transition flex items-center gap-2">
    <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20"
      fill="currentColor">
      <path d="M2 6a2 2 0 012-2h6a2 2 0 012 2v8a2 2 0 01-2 2H4a2 2 0 01-2-2V6zM
14 6a2 2 0 012-2h2a2 2 0 012 2v8a2 2 0 01-2 2h-2a2 2 0 01-2-2V6z" />
    </svg>
    Video auswählen
  </label>
  <video id="VideoPlayer" controls class="mt-6 w-full max-w-2xl rounded-lg hidd
en"></video>
  <p id="dauerGeschnittenesVideo" class="mt-3 text-gray-400 hidden"></p>
  <button id="geschnittenesVideoPlayBtn"
    class="mt-4 bg-purple-600 hover:bg-purple-700 text-white font-medium
    py-2 px-4 rounded-lg transition flex items-center gap-2 hidden">
    <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20"
      fill="currentColor">
      <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000 16zM9.555 7.16
8A1 1 0 008 8v4a1 1 0 001.555.832l3-2a1 1 0 000-1.664l-3-2z" clip-rule="evenodd" />
    </svg>
    Geschnittenes Video abspielen
  </button>
</div>

<div id="restContainer" class="hidden space-y-8">
  <div id="SchnittContainer" class="space-y-6"></div>

  <button id="addNewCutBtn"
    class="w-full bg-gray-700 hover:bg-gray-600 text-white font-medium py
-3 rounded-lg transition flex items-center justify-center gap-2">
    <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20"
      fill="currentColor">
      <path fill-rule="evenodd" d="M10 5a1 1 0 011 1v3h3a1 1 0 110 2h-3v3a1 1 0
11-2 0v-3H6a1 1 0 110-2h3V6a1 1 0 011-1z" clip-rule="evenodd" />
    </svg>
    Schnitt hinzufügen
  </button>


```

```

        <div class="Beschreibung">
            <label for="BeschreibungInput" class="block text-sm font-medium text-gray-300 mb-2">Beschreibung:</label>
            <input type="text" id="BeschreibungInput"
                class="w-full bg-gray-700 border border-gray-600 rounded-lg px-4 py-3 text-white focus:outline-none focus:ring-2 focus:ring-blue-500"
                placeholder="Beschreiben Sie, was Sie geschnitten haben"/>
        </div>

        <button id="AbsendeBtn"
            class="w-full bg-green-600 hover:bg-green-700 text-white font-medium py-3 rounded-lg transition flex items-center justify-center gap-2">
            <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20"
            fill="currentColor">
                <path fill-rule="evenodd" d="M16.707 5.293a1 1 0 010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1 0 011.414 0z" clip-rule="evenodd" />
            </svg>
            Video absenden
        </button>
    </div>
</div>

<script src="static/script.js"></script>
</body>
</html>

```

script.js (Anpassungen)

javascript

```

// In der addNewCutBtn.addEventListener-Funktion:
cutSection.innerHTML = `
    <div class="flex justify-between items-center mb-3">
        <h3 class="text-lg font-medium">Schnitt ${cutSectionCount}</h3>
        <button class="deleteCutBtn bg-red-600 hover:bg-red-700 text-white p-2 rounded-full transition" style="margin-right: 20px">
            <svg xmlns="http://www.w3.org/2000/svg" class="h-4 w-4" viewBox="0 0 20 20"
            fill="currentColor">
                <path fill-rule="evenodd" d="M4.293 4.293a1 1 0 011.414 0L10 8.586l4.293-4.293a1 1 0 011.414 1.414L11.414 10L4.293 4.293a1 1 0 01-1.414 1.414L10 11.414l-4.293 4.293a1 1 0 01-1.414-1.414L8.586 10 4.293 5.707a1 1 0 010-1.414z" clip-rule="evenodd" />
            </svg>
        </button>
    </div>

```

```

</div>
<div class="space-y-4">
  <div>
    <label class="block text-sm text-gray-400 mb-1">Startzeit</label>
    <div class="flex items-center gap-4">
      <input type="range" class="startRange w-full h-2 bg-gray-700 rounded-lg appearance-none cursor-pointer" min="0" step="0.01" value="0"/>
      <input type="text" class="startInput bg-gray-700 border border-gray-600 rounded px-3 py-2 text-white w-24 text-center" maxlength="8" value="00:00.00" />
    </div>
  </div>
  <div>
    <label class="block text-sm text-gray-400 mb-1">Endzeit</label>
    <div class="flex items-center gap-4">
      <input type="range" class="endRange w-full h-2 bg-gray-700 rounded-lg appearance-none cursor-pointer" min="0" step="0.01" value="0"/>
      <input type="text" class="endInput bg-gray-700 border border-gray-600 rounded px-3 py-2 text-white w-24 text-center" maxlength="8" value="00:00.01" />
    </div>
  </div>
</div>
`;

// In der popup-Funktion:
overlay.className = "fixed inset-0 bg-black bg-opacity-70 flex items-center justify-center z-50";
popup.className = "bg-gray-800 rounded-xl p-6 max-w-md w-full shadow-2xl";
message.className = "text-white mb-4 text-center";
okButton.className = "w-full bg-blue-600 hover:bg-blue-700 text-white font-medium py-2 px-4 rounded-lg transition mt-4";

// In der resetLayout-Funktion:
// Änderung für das Label
videoInputLabel.style.display = "flex"; // Ändern auf flex wegen des Icons

// In der videoInput.addEventListener-Funktion:
// Änderung für das Label
videoInputLabel.style.display = "none"; // Flex wird zu none

```

Wichtige Änderungen:

1. Modernes Dark Theme:

- Dunkler Hintergrund (gray-900)
- Karten mit gray-800 und Schatten

- Akzentfarben (blau, lila) für Interaktionen
- Sanfte Übergänge bei Hover-Effekten

2. Icons hinzugefügt:

- Videoauswahl: Videokamera-Icon
- Schnitt hinzufügen: Plus-Icon
- Absenden: Häkchen-Icon
- Löschen: X-Icon
- Play: Play-Icon

3. Verbesserte UI-Komponenten:

- Buttons mit Icons und Hover-Effekten
- Eingabefelder mit dunklem Design
- Slider mit blauem Thumb
- Abgerundete Ecken für alle Elemente
- Responsives Layout

4. Dynamische Elemente gestaltet:

- Schnittabschnitte als Karten mit Schatten
- Popup als modales Overlay
- Bessere Anordnung der Range-Slider und Textfelder

5. Typografie:

- Inter Font für modernes Aussehen
- Lesbare Kontraste
- Gradient für den Titel

6. JavaScript-Anpassungen:

- CSS-Klassen für dynamische Elemente hinzugefügt
- Display-Anpassungen für Flex-Layout
- Styling der Range-Slider

Das Design ist professionell, modern und funktional, ohne zu überladen. Alle Funktionalitäten bleiben erhalten, während das UI deutlich aufgewertet wurde.