<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SWEET POINT - Video Platform</title>

<!-- Tailwind CSS for styling -->

<script src="https://cdn.tailwindcss.com"></script>

<!-- Google Fonts: Inter -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap" rel="stylesheet">

<!-- Custom Styles -->

<style>

body {

font-family: 'Inter', sans-serif;

}

/\* Custom scrollbar for a more modern look \*/

::-webkit-scrollbar {

width: 8px;

}

::-webkit-scrollbar-track {

background: #1e293b;

}

::-webkit-scrollbar-thumb {

background: #475569;

border-radius: 10px;

}

::-webkit-scrollbar-thumb:hover {

background: #64748b;

}

</style>

</head>

<body class="bg-slate-900 text-white antialiased">

<!-- Header -->

<header class="bg-slate-800/50 backdrop-blur-sm sticky top-0 z-50 shadow-md">

<div class="container mx-auto px-4 sm:px-6 lg:px-8">

<div class="flex items-center justify-between h-16">

<h1 class="text-2xl font-bold text-rose-500 tracking-wider">SWEET POINT</h1>

<div class="flex items-center space-x-4">

<input type="text" placeholder="Search..." class="hidden md:block bg-slate-700 border border-slate-600 rounded-full px-4 py-2 text-sm focus:outline-none focus:ring-2 focus:ring-rose-500 transition duration-300">

<svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 text-slate-400" fill="none" viewBox="0 0 24 24" stroke="currentColor">

<path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M15 17h5l-1.405-1.405A2.032 2.032 0 0118 14.158V11a6 6 0 00-5-5.917V5a2 2 0 10-4 0v.083A6 6 0 004 11v3.159c0 .538-.214 1.055-.595 1.436L2 17h5m6 0v1a3 3 0 11-6 0v-1m6 0H9" />

</svg>

</div>

</div>

</div>

</header>

<!-- Main Content -->

<main class="container mx-auto p-4 sm:p-6 lg:p-8">

<div class="grid grid-cols-1 lg:grid-cols-3 gap-8">

<!-- Video Player and Details (Left/Main Column) -->

<div class="lg:col-span-2">

<!-- Video Player -->

<div class="bg-black rounded-xl overflow-hidden aspect-video shadow-2xl shadow-rose-500/10">

<video id="main-video" class="w-full h-full" controls poster="https://placehold.co/1280x720/000000/FFFFFF?text=SWEET+POINT" src="http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/BigBuckBunny.mp4">

Your browser does not support the video tag.

</video>

</div>

<!-- Video Info -->

<div class="mt-6">

<h2 id="video-title" class="text-2xl md:text-3xl font-bold leading-tight">Big Buck Bunny: A Tale of a Gentle Giant</h2>

<div class="flex items-center justify-between mt-4 flex-wrap gap-4">

<!-- Channel Info -->

<div class="flex items-center space-x-3">

<img src="https://placehold.co/48x48/ec4899/ffffff?text=B" class="rounded-full" alt="Channel Avatar">

<div>

<p id="channel-name" class="font-semibold">Blender Foundation</p>

<p class="text-sm text-slate-400">1.2M subscribers</p>

</div>

</div>

<!-- Action Buttons -->

<div class="flex items-center space-x-3">

<button id="like-btn" class="flex items-center space-x-2 bg-slate-800 hover:bg-slate-700 transition-colors duration-300 px-4 py-2 rounded-full font-semibold">

<svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20" fill="currentColor">

<path d="M2 10.5a1.5 1.5 0 113 0v6a1.5 1.5 0 01-3 0v-6zM6 10.333v5.43a2 2 0 001.106 1.787l.25.125a2 2 0 002.29-1.584l.955-4.773a2 2 0 00-1.992-2.363H8.5l.002-.001a.498.498 0 01.352-.146l1.54-.854a2 2 0 00.818-2.177l-.45-1.8a2 2 0 00-1.95-1.543H7.5a2 2 0 00-2 2v2.283l-.623 2.492a.5.5 0 00.48 1.125H6z" />

</svg>

<span>Like</span>

<span id="like-count" class="text-sm text-slate-400 ml-1">12K</span>

</button>

<button id="share-btn" class="flex items-center space-x-2 bg-slate-800 hover:bg-slate-700 transition-colors duration-300 px-4 py-2 rounded-full font-semibold">

<svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5" viewBox="0 0 20 20" fill="currentColor">

<path d="M15 8a3 3 0 10-2.977-2.63l-4.94 2.47a3 3 0 100 4.319l4.94 2.47a3 3 0 10.895-1.789l-4.94-2.47a3.027 3.027 0 000-.74l4.94-2.47C13.456 7.68 14.19 8 15 8z" />

</svg>

<span>Share</span>

</button>

</div>

</div>

</div>

<!-- Video Description -->

<div class="mt-6 bg-slate-800 p-4 rounded-xl">

<p id="video-description" class="text-slate-300 text-sm leading-relaxed">

A large and lovable rabbit, "Big Buck Bunny," enjoys a peaceful life in the forest. His day is disrupted by three mischievous rodents: Frank the flying squirrel, Rinky the chinchilla, and Gamera the turtle. They bully smaller creatures and eventually provoke the gentle giant by killing a butterfly he befriended. Angered, Bunny sets aside his gentle nature and devises a series of clever traps to exact his revenge.

</p>

</div>

</div>

<!-- Up Next (Right/Side Column) -->

<div class="lg:col-span-1">

<h3 class="text-xl font-bold mb-4">Up Next</h3>

<div id="up-next-list" class="space-y-4">

<!-- Video items will be injected here by JavaScript -->

</div>

</div>

</div>

</main>

<!-- Floating message for 'Share' action -->

<div id="share-message" class="fixed bottom-5 right-5 bg-rose-500 text-white px-6 py-3 rounded-lg shadow-lg opacity-0 transform translate-y-10 transition-all duration-500">

Link copied to clipboard!

</div>

<script>

// --- DOM Elements ---

const mainVideo = document.getElementById('main-video');

const videoTitle = document.getElementById('video-title');

const channelName = document.getElementById('channel-name');

const videoDescription = document.getElementById('video-description');

const likeBtn = document.getElementById('like-btn');

const likeCountSpan = document.getElementById('like-count');

const shareBtn = document.getElementById('share-btn');

const shareMessage = document.getElementById('share-message');

const upNextList = document.getElementById('up-next-list');

// --- Data for Videos ---

const videoData = [

{

id: 1,

title: 'Elephants Dream: The First Open Movie',

channel: 'Blender Foundation',

description: 'Elephants Dream is a 2006 computer-animated science fiction fantasy short film that was produced by the Blender Foundation using exclusively free and open-source software.',

videoSrc: 'http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ElephantsDream.mp4',

posterSrc: 'https://placehold.co/320x180/1e293b/ffffff?text=Elephants+Dream',

likes: 8500

},

{

id: 2,

title: 'For Bigger Blazes: More Action!',

channel: 'Action Flix',

description: 'Experience high-octane action in this short clip from the "For Bigger Blazes" series. Perfect for testing your sound system.',

videoSrc: 'http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/ForBiggerBlazes.mp4',

posterSrc: 'https://placehold.co/320x180/1e293b/ffffff?text=Bigger+Blazes',

likes: 22000

},

{

id: 3,

title: 'Sintel: A CGI Fantasy Adventure',

channel: 'Blender Foundation',

description: 'Sintel is a 2010 computer-animated fantasy short film. The film follows a girl named Sintel who is searching for her pet baby dragon, Scales.',

videoSrc: 'http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/Sintel.mp4',

posterSrc: 'https://placehold.co/320x180/1e293b/ffffff?text=Sintel',

likes: 45000

},

{

id: 4,

title: 'Tears of Steel: Live-Action/CGI Sci-Fi',

channel: 'Blender Foundation',

description: 'Tears of Steel is a 2012 science fiction short film by producer Ton Roosendaal and director/writer Ian Hubert. It centers on a group of warriors and scientists who gather at the Oude Kerk in a future Amsterdam.',

videoSrc: 'http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/TearsOfSteel.mp4',

posterSrc: 'https://placehold.co/320x180/1e293b/ffffff?text=Tears+of+Steel',

likes: 31000

}

];

let currentLikes = 12000;

let liked = false;

// --- Functions ---

/\*\*

\* Formats large numbers into a more readable format (e.g., 12000 -> 12K)

\* @param {number} num - The number to format.

\* @returns {string} The formatted number string.

\*/

function formatNumber(num) {

if (num >= 1000000) {

return (num / 1000000).toFixed(1).replace(/\.0$/, '') + 'M';

}

if (num >= 1000) {

return (num / 1000).toFixed(1).replace(/\.0$/, '') + 'K';

}

return num;

}

/\*\*

\* Populates the "Up Next" list with video items.

\*/

function populateUpNext() {

upNextList.innerHTML = ''; // Clear existing list

videoData.forEach(video => {

const videoItem = document.createElement('div');

videoItem.className = 'flex items-start space-x-4 cursor-pointer group';

videoItem.innerHTML = `

<div class="w-2/5 flex-shrink-0">

<img src="${video.posterSrc}" class="rounded-lg aspect-video object-cover group-hover:scale-105 transition-transform duration-300" alt="${video.title}">

</div>

<div class="w-3/5">

<h4 class="font-semibold text-sm leading-tight group-hover:text-rose-400 transition-colors">${video.title}</h4>

<p class="text-xs text-slate-400 mt-1">${video.channel}</p>

</div>

`;

// Add click listener to change the main video

videoItem.addEventListener('click', () => loadVideo(video));

upNextList.appendChild(videoItem);

});

}

/\*\*

\* Loads a new video into the main player and updates the details.

\* @param {object} video - The video object to load.

\*/

function loadVideo(video) {

mainVideo.src = video.videoSrc;

mainVideo.poster = video.posterSrc;

videoTitle.textContent = video.title;

channelName.textContent = video.channel;

videoDescription.textContent = video.description;

currentLikes = video.likes;

likeCountSpan.textContent = formatNumber(currentLikes);

liked = false; // Reset like status for new video

likeBtn.classList.remove('text-rose-500'); // Reset like button color

mainVideo.play();

window.scrollTo(0, 0); // Scroll to top

}

/\*\*

\* Handles the like button click event.

\*/

function handleLike() {

if (liked) {

currentLikes--;

likeBtn.classList.remove('text-rose-500');

} else {

currentLikes++;

likeBtn.classList.add('text-rose-500');

}

liked = !liked;

likeCountSpan.textContent = formatNumber(currentLikes);

}

/\*\*

\* Handles the share button click event.

\*/

function handleShare() {

// Create a temporary textarea to copy the text

const tempTextArea = document.createElement('textarea');

tempTextArea.value = window.location.href; // You can change this to a specific video URL

document.body.appendChild(tempTextArea);

tempTextArea.select();

document.execCommand('copy');

document.body.removeChild(tempTextArea);

// Show confirmation message

shareMessage.classList.remove('opacity-0', 'translate-y-10');

shareMessage.classList.add('opacity-100', 'translate-y-0');

// Hide the message after 3 seconds

setTimeout(() => {

shareMessage.classList.remove('opacity-100', 'translate-y-0');

shareMessage.classList.add('opacity-0', 'translate-y-10');

}, 3000);

}

// --- Event Listeners ---

likeBtn.addEventListener('click', handleLike);

shareBtn.addEventListener('click', handleShare);

// --- Initial Load ---

document.addEventListener('DOMContentLoaded', () => {

populateUpNext();

});

</script>

</body>

</html>