```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>StudyTube - Retro Educational Hub</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=VT323&family=Press+Start+2P&display=swap');

        :root {
            --primary-color: #33ff33;
            --secondary-color: #00cc00;
            --bg-color: #0a0a0a;
            --card-bg: #111111;
            --text-color: #33ff33;
            --accent-color: #ffcc00;
            --danger-color: #ff3333;
            --border-color: #33ff33;
            --download-color: #00ffff;
        }

        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: 'VT323', monospace;
            background-color: var(--bg-color);
            color: var(--text-color);
            line-height: 1.6;
            overflow-x: hidden;
            position: relative;
        }

        body::before {
            content: " ";
            display: block;
            position: fixed;
            top: 0;
            left: 0;
            bottom: 0;
            right: 0;
            background: linear-gradient(rgba(18, 16, 16, 0) 50%, rgba(0, 0, 0, 0.25) 50%), linear-gradient(90deg, rgba(255, 0, 0, 0.06), rgba(0, 255, 0, 0.02), rgba(0, 0, 255, 0.06));
            z-index: 2;
            background-size: 100% 2px, 3px 100%;
            pointer-events: none;
        }

        .container {
            max-width: 1400px;
            margin: 0 auto;
            padding: 0 20px;
            position: relative;
            z-index: 1;
        }

        /* Header */
        header {
            background-color: rgba(17, 17, 17, 0.9);
            border-bottom: 2px solid var(--border-color);
            padding: 15px 0;
            position: sticky;
            top: 0;
            z-index: 100;
            box-shadow: 0 0 10px rgba(51, 255, 51, 0.3);
        }

        .header-content {
            display: flex;
            justify-content: space-between;
            align-items: center;
        }

        .logo {
            font-family: 'Press Start 2P', cursive;
            font-size: 24px;
            color: var(--accent-color);
            text-shadow: 2px 2px 0 rgba(0, 0, 0, 0.5);
            display: flex;
            align-items: center;
        }
```

```css
.logo i {
    margin-right: 10px;
    font-size: 30px;
}

.search-bar {
    flex-grow: 1;
    max-width: 600px;
    margin: 0 20px;
    position: relative;
}

.search-bar input {
    width: 100%;
    padding: 10px 40px 10px 15px;
    background-color: rgba(51, 255, 51, 0.1);
    border: 2px solid var(--border-color);
    color: var(--text-color);
    font-family: 'VT323', monospace;
    font-size: 20px;
    border-radius: 0;
    outline: none;
}

.search-bar button {
    position: absolute;
    right: 5px;
    top: 50%;
    transform: translateY(-50%);
    background: none;
    border: none;
    color: var(--text-color);
    cursor: pointer;
    padding: 5px 10px;
}

.user-actions {
    display: flex;
    gap: 15px;
    align-items: center;
}

.btn {
    padding: 8px 15px;
    background-color: transparent;
    border: 2px solid var(--border-color);
    color: var(--text-color);
    cursor: pointer;
    font-family: 'VT323', monospace;
    font-size: 18px;
    transition: all 0.3s;
    position: relative;
    overflow: hidden;
}

.btn:hover {
    background-color: rgba(51, 255, 51, 0.2);
    box-shadow: 0 0 10px rgba(51, 255, 51, 0.5);
}

.btn-primary {
    background-color: var(--secondary-color);
    color: var(--bg-color);
}

.btn-danger {
    border-color: var(--danger-color);
    color: var(--danger-color);
    font-weight: bold;
    animation: pulse 2s infinite;
}

@keyframes pulse {
    0% { box-shadow: 0 0 0 0 rgba(255, 51, 51, 0.7); }
    70% { box-shadow: 0 0 0 10px rgba(255, 51, 51, 0); }
    100% { box-shadow: 0 0 0 0 rgba(255, 51, 51, 0); }
}

.btn-danger:hover {
    background-color: rgba(255, 51, 51, 0.2);
    box-shadow: 0 0 10px rgba(255, 51, 51, 0.5);
}

/* Fixed Exit Button */
.fixed-exit {
    position: fixed;
    bottom: 30px;
    left: 30px;
    z-index: 1000;
    background-color: var(--danger-color);
    color: var(--bg-color);
    padding: 15px 20px;
    border: 2px solid var(--danger-color);
    border-radius: 50%;
    font-size: 24px;
    cursor: pointer;
    transition: all 0.3s;
    box-shadow: 0 4px 15px rgba(255, 51, 51, 0.5);
    animation: float 3s ease-in-out infinite;
}

@keyframes float {
    0%, 100% { transform: translateY(0px); }
    50% { transform: translateY(-10px); }
}

.fixed-exit:hover {
    transform: scale(1.1);
    box-shadow: 0 6px 20px rgba(255, 51, 51, 0.7);
}

/* Navigation */
nav {
    background-color: rgba(17, 17, 17, 0.8);
    padding: 15px 0;
    border-bottom: 1px solid var(--border-color);
}

.nav-tabs {
    display: flex;
    gap: 20px;
    list-style: none;
}

.nav-tab {
    padding: 8px 15px;
    cursor: pointer;
    font-size: 20px;
    position: relative;
    transition: all 0.3s;
}
```

```
.nav-tab:hover {
    color: var(--accent-color);
}

.nav-tab.active {
    color: var(--accent-color);
}

.nav-tab.active::after {
    content: "";
    position: absolute;
    bottom: -15px;
    left: 0;
    width: 100%;
    height: 3px;
    background-color: var(--accent-color);
}

/* Main Content */
main {
    padding: 30px 0;
    min-height: calc(100vh - 200px);
}

.content-section {
    display: none;
}

.content-section.active {
    display: block;
}

.section-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 20px;
    padding-bottom: 10px;
    border-bottom: 1px solid var(--border-color);
}

.section-title {
    font-family: 'Press Start 2P', cursive;
    font-size: 20px;
    color: var(--accent-color);
}

.filter-options {
    display: flex;
    gap: 10px;
}

.filter-btn {
    padding: 5px 10px;
    background-color: transparent;
    border: 1px solid var(--border-color);
    color: var(--text-color);
    cursor: pointer;
    font-family: 'VT323', monospace;
    font-size: 18px;
}

.filter-btn:hover {
    background-color: rgba(51, 255, 51, 0.1);
}

/* Content Grid */
.content-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 25px;
}

.content-card {
    background-color: var(--card-bg);
    border: 2px solid var(--border-color);
    overflow: hidden;
    transition: all 0.3s;
    cursor: pointer;
}

.content-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 20px rgba(51, 255, 51, 0.3);
}

.content-thumbnail {
    position: relative;
    height: 180px;
    background-color: #0a0a0a;
    overflow: hidden;
}

.content-thumbnail img {
    width: 100%;
    height: 100%;
    object-fit: cover;
}

.content-duration {
    position: absolute;
    bottom: 10px;
    right: 10px;
    background-color: rgba(0, 0, 0, 0.8);
    color: var(--text-color);
    padding: 2px 5px;
    font-size: 16px;
}

.content-info {
    padding: 15px;
}

.content-title {
    font-size: 20px;
    margin-bottom: 5px;
    overflow: hidden;
    text-overflow: ellipsis;
    white-space: nowrap;
}

.content-meta {
    display: flex;
    justify-content: space-between;
    margin-bottom: 10px;
    font-size: 16px;
    color: rgba(51, 255, 51, 0.7);
}

.content-author {
    display: flex;
    align-items: center;
    gap: 5px;
}

.author-avatar {
    width: 24px;
    height: 24px;
    border-radius: 50%;
    background-color: var(--secondary-color);
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 14px;
    color: var(--bg-color);
}

.content-stats {
    display: flex;
    gap: 15px;
    font-size: 16px;
}

.content-actions {
    display: flex;
    justify-content: space-between;
    padding: 10px 15px;
    border-top: 1px solid var(--border-color);
}

.action-btn {
    background: none;
    border: none;
    color: var(--text-color);
    cursor: pointer;
    font-size: 18px;
    display: flex;
    align-items: center;
    gap: 5px;
    transition: all 0.3s;
}

.action-btn:hover {
    color: var(--accent-color);
}
```

```
.action-btn.liked {
    color: var(--danger-color);
}

.action-btn.download-btn {
    color: var(--download-color);
}

.action-btn.download-btn:hover {
    color: #00cccc;
    box-shadow: 0 0 5px rgba(0, 255, 255, 0.5);
}

/* Upload Modal */
.modal {
    display: none;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.8);
    z-index: 1000;
    align-items: center;
    justify-content: center;
}

.modal.active {
    display: flex;
}

.modal-content {
    background-color: var(--card-bg);
    border: 2px solid var(--border-color);
    padding: 30px;
    width: 90%;
    max-width: 600px;
    position: relative;
}

.modal-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 20px;
    padding-bottom: 10px;
    border-bottom: 1px solid var(--border-color);
}

.modal-title {
    font-family: 'Press Start 2P', cursive;
    font-size: 18px;
    color: var(--accent-color);
}

.close-modal {
    background: none;
    border: none;
    color: var(--danger-color);
    font-size: 24px;
    cursor: pointer;
}

.form-group {
    margin-bottom: 20px;
}

.form-group label {
    display: block;
    margin-bottom: 5px;
    font-size: 18px;
}

.form-group input,
.form-group textarea,
.form-group select {
    width: 100%;
    padding: 10px;
    background-color: rgba(51, 255, 51, 0.1);
    border: 1px solid var(--border-color);
    color: var(--text-color);
    font-family: 'VT323', monospace;
    font-size: 18px;
    outline: none;
}

.form-group textarea {
    resize: vertical;
    min-height: 100px;
}

.file-upload {
    position: relative;
    display: inline-block;
    cursor: pointer;
    width: 100%;
}

.file-upload input[type=file] {
    position: absolute;
    left: -9999px;
}

.file-upload-label {
    display: block;
    padding: 10px;
    background-color: rgba(51, 255, 51, 0.1);
    border: 1px dashed var(--border-color);
    text-align: center;
    font-size: 18px;
}

.form-actions {
    display: flex;
    justify-content: flex-end;
    gap: 10px;
}

/* Video Player Modal */
.video-player {
    display: none;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.9);
    z-index: 1000;
    align-items: center;
    justify-content: center;
}
```

```
.video-player.active {
   display: flex;
}

.player-container {
   width: 90%;
   max-width: 900px;
   height: 90vh;
   display: flex;
   flex-direction: column;
}

.player-header {
   display: flex;
   justify-content: space-between;
   align-items: center;
   padding: 10px;
   background-color: var(--card-bg);
   border: 1px solid var(--border-color);
   border-bottom: none;
}

.player-title {
   font-size: 20px;
}

.close-player {
   background: none;
   border: none;
   color: var(--danger-color);
   font-size: 24px;
   cursor: pointer;
}

.player-video {
   width: 100%;
   height: 300px;
   background-color: #000;
   border: 1px solid var(--border-color);
   display: flex;
   align-items: center;
   justify-content: center;
   font-size: 24px;
   color: rgba(51, 255, 51, 0.5);
}

.player-info {
   padding: 15px;
   background-color: var(--card-bg);
   border: 1px solid var(--border-color);
   border-top: none;
   flex-grow: 1;
   overflow-y: auto;
}

.player-stats {
   display: flex;
   gap: 20px;
   margin-bottom: 15px;
}

.player-actions {
   display: flex;
   gap: 15px;
   margin-bottom: 15px;
   flex-wrap: wrap;
}

.player-action-btn {
   background: none;
   border: 1px solid var(--border-color);
   color: var(--text-color);
   padding: 5px 10px;
   cursor: pointer;
   font-size: 18px;
   display: flex;
   align-items: center;
   gap: 5px;
}

.player-action-btn:hover {
   background-color: rgba(51, 255, 51, 0.1);
}

.player-action-btn.liked {
   color: var(--danger-color);
   border-color: var(--danger-color);
}
```

```
.player-action-btn.download-btn {
   color: var(--download-color);
   border-color: var(--download-color);
}

.player-action-btn.download-btn:hover {
   background-color: rgba(0, 255, 255, 0.1);
   box-shadow: 0 0 10px rgba(0, 255, 255, 0.3);
}

.player-description {
   margin-top: 15px;
   padding-top: 15px;
   border-top: 1px solid var(--border-color);
   font-size: 18px;
}

.comments-section {
   margin-top: 20px;
   padding-top: 15px;
   border-top: 1px solid var(--border-color);
}

.comments-header {
   display: flex;
   justify-content: space-between;
   align-items: center;
   margin-bottom: 15px;
}

.comments-count {
   font-size: 20px;
   color: var(--accent-color);
}

.sort-comments {
   padding: 5px 10px;
   background-color: rgba(51, 255, 51, 0.1);
   border: 1px solid var(--border-color);
   color: var(--text-color);
   font-family: 'VT323', monospace;
   font-size: 16px;
   cursor: pointer;
```

```
}

.comment-form {
    display: flex;
    gap: 10px;
    margin-bottom: 20px;
}

.comment-input {
    flex-grow: 1;
    padding: 10px;
    background-color: rgba(51, 255, 51, 0.1);
    border: 1px solid var(--border-color);
    color: var(--text-color);
    font-family: 'VT323', monospace;
    font-size: 18px;
    outline: none;
}

.comment-submit {
    padding: 10px 15px;
    background-color: var(--secondary-color);
    border: none;
    color: var(--bg-color);
    cursor: pointer;
    font-family: 'VT323', monospace;
    font-size: 18px;
}

.comments-container {
    max-height: 300px;
    overflow-y: auto;
    padding-right: 10px;
    border: 1px solid rgba(51, 255, 51, 0.2);
    padding: 10px;
    background-color: rgba(0, 0, 0, 0.3);
}

.comments-container::-webkit-scrollbar {
    width: 8px;
}

.comments-container::-webkit-scrollbar-track {
    background: rgba(51, 255, 51, 0.1);
```

```
}

.comments-container::-webkit-scrollbar-thumb {
   background: var(--border-color);
   border-radius: 4px;
}

.comments-container::-webkit-scrollbar-thumb:hover {
   background: var(--secondary-color);
}

.comment {
   display: flex;
   gap: 10px;
   margin-bottom: 15px;
   padding-bottom: 15px;
   border-bottom: 1px solid rgba(51, 255, 51, 0.2);
}

.comment:last-child {
   border-bottom: none;
}

.comment-avatar {
   width: 40px;
   height: 40px;
   border-radius: 50%;
   background-color: var(--secondary-color);
   display: flex;
   align-items: center;
   justify-content: center;
   font-size: 20px;
   color: var(--bg-color);
   flex-shrink: 0;
}

.comment-content {
   flex-grow: 1;
}

.comment-author {
   font-size: 18px;
   margin-bottom: 5px;
   color: var(--accent-color);
```

```
}

.comment-text {
  font-size: 18px;
  margin-bottom: 5px;
  line-height: 1.4;
}

.comment-actions {
  display: flex;
  gap: 15px;
  font-size: 16px;
  color: rgba(51, 255, 51, 0.7);
}

.comment-action {
  cursor: pointer;
  transition: color 0.3s;
}

.comment-action:hover {
  color: var(--text-color);
}

.load-more-comments {
  text-align: center;
  margin-top: 15px;
}

.load-more-btn {
  padding: 8px 20px;
  background-color: transparent;
  border: 1px solid var(--border-color);
  color: var(--text-color);
  cursor: pointer;
  font-family: 'VT323', monospace;
  font-size: 18px;
  transition: all 0.3s;
}

.load-more-btn:hover {
  background-color: rgba(51, 255, 51, 0.1);
}
```

```
/* Download Progress Modal */
.download-modal {
    display: none;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.8);
    z-index: 2000;
    align-items: center;
    justify-content: center;
}

.download-modal.active {
    display: flex;
}

.download-content {
    background-color: var(--card-bg);
    border: 2px solid var(--download-color);
    padding: 30px;
    text-align: center;
    position: relative;
}

.download-title {
    font-family: 'Press Start 2P', cursive;
    font-size: 18px;
    color: var(--download-color);
    margin-bottom: 20px;
}

.download-progress {
    width: 300px;
    height: 30px;
    background-color: rgba(0, 0, 0, 0.5);
    border: 1px solid var(--border-color);
    margin: 20px 0;
    position: relative;
    overflow: hidden;
}

.download-progress-bar {
    height: 100%;
    background: linear-gradient(90deg, var(--download-color), var(--primary-color));
    width: 0%;
    transition: width 0.3s;
}

.download-percentage {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    font-size: 18px;
    color: var(--text-color);
}

.download-status {
    font-size: 20px;
    margin-top: 15px;
}

/* Notification Toast */
.toast {
    position: fixed;
    bottom: 20px;
    right: 20px;
    background-color: var(--card-bg);
    border: 1px solid var(--border-color);
    padding: 15px 20px;
    display: flex;
    align-items: center;
    gap: 10px;
    transform: translateX(400px);
    transition: transform 0.3s;
    z-index: 1001;
}

.toast.show {
    transform: translateX(0);
}

.toast-icon {
    font-size: 20px;
}
```

```
.toast-message {
   font-size: 18px;
}

/* Footer */
footer {
   background-color: rgba(17, 17, 17, 0.9);
   border-top: 2px solid var(--border-color);
   padding: 20px 0;
   text-align: center;
   font-size: 18px;
   color: rgba(51, 255, 51, 0.7);
}

/* Responsive Design */
@media (max-width: 768px) {
   .header-content {
      flex-direction: column;
      gap: 15px;
   }

   .search-bar {
      width: 100%;
      margin: 0;
   }

   .nav-tabs {
      overflow-x: auto;
      padding-bottom: 5px;
   }

   .content-grid {
      grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
   }

   .player-video {
      height: 200px;
   }

   .fixed-exit {
      bottom: 20px;
      left: 20px;
      padding: 12px 16px;
      font-size: 20px;
```

```
            }
        }
    </style>
</head>
<body>
    <header>
        <div class="container">
            <div class="header-content">
                <div class="logo">
                    <i class="fas fa-graduation-cap"></i>
                    StudyTube
                </div>
                <div class="search-bar">
                    <input type="text" id="searchInput" placeholder="Search for study materials...">
                    <button id="searchBtn"><i class="fas fa-search"></i></button>
                </div>
                <div class="user-actions">
                    <button class="btn" id="uploadBtn"><i class="fas fa-upload"></i> Upload</button>
                    <button class="btn" id="notificationBtn"><i class="fas fa-bell"></i></button>
                    <button class="btn" id="profileBtn"><i class="fas fa-user-circle"></i></button>
                    <button class="btn btn-danger" id="headerExitBtn"><i class="fas
fa-sign-out-alt"></i> Exit</button>
                </div>
            </div>
        </div>
    </header>

    <!-- Fixed Exit Button -->
    <button class="fixed-exit" id="fixedExitBtn" title="Exit StudyTube">
        <i class="fas fa-power-off"></i>
    </button>

    <nav>
        <div class="container">
            <ul class="nav-tabs">
                <li class="nav-tab active" data-tab="trending">Trending</li>
                <li class="nav-tab" data-tab="subscriptions">Subscriptions</li>
                <li class="nav-tab" data-tab="library">Library</li>
                <li class="nav-tab" data-tab="my-videos">My Videos</li>
            </ul>
        </div>
    </nav>

    <main>
```

```html
<div class="container">
    <!-- Trending Section -->
    <section class="content-section active" id="trending">
        <div class="section-header">
            <h2 class="section-title">TRENDING NOW</h2>
            <div class="filter-options">
                <button class="filter-btn">All Subjects</button>
                <button class="filter-btn">Math</button>
                <button class="filter-btn">Science</button>
                <button class="filter-btn">History</button>
                <button class="filter-btn">Languages</button>
            </div>
        </div>
        <div class="content-grid" id="trendingGrid">
            <!-- Content will be dynamically added here -->
        </div>
    </section>

    <!-- Subscriptions Section -->
    <section class="content-section" id="subscriptions">
        <div class="section-header">
            <h2 class="section-title">SUBSCRIPTIONS</h2>
            <div class="filter-options">
                <button class="filter-btn">All</button>
                <button class="filter-btn">Today</button>
                <button class="filter-btn">This Week</button>
                <button class="filter-btn">This Month</button>
            </div>
        </div>
        <div class="content-grid" id="subscriptionsGrid">
            <!-- Content will be dynamically added here -->
        </div>
    </section>

    <!-- Library Section -->
    <section class="content-section" id="library">
        <div class="section-header">
            <h2 class="section-title">YOUR LIBRARY</h2>
            <div class="filter-options">
                <button class="filter-btn">All</button>
                <button class="filter-btn">Videos</button>
                <button class="filter-btn">Images</button>
                <button class="filter-btn">Recently Viewed</button>
            </div>
```

```
                </div>
                <div class="content-grid" id="libraryGrid">
                    <!-- Content will be dynamically added here -->
                </div>
            </section>

            <!-- My Videos Section -->
            <section class="content-section" id="my-videos">
                <div class="section-header">
                    <h2 class="section-title">MY CONTENT</h2>
                    <div class="filter-options">
                        <button class="filter-btn">All</button>
                        <button class="filter-btn">Public</button>
                        <button class="filter-btn">Private</button>
                        <button class="filter-btn">Drafts</button>
                    </div>
                </div>
                <div class="content-grid" id="myVideosGrid">
                    <!-- Content will be dynamically added here -->
                </div>
            </section>
        </div>
    </main>

    <!-- Upload Modal -->
    <div class="modal" id="uploadModal">
        <div class="modal-content">
            <div class="modal-header">
                <h3 class="modal-title">UPLOAD CONTENT</h3>
                <button class="close-modal" id="closeUploadModal"><i class="fas
fa-times"></i></button>
            </div>
            <form id="uploadForm">
                <div class="form-group">
                    <label for="contentType">Content Type</label>
                    <select id="contentType">
                        <option value="video">Video</option>
                        <option value="image">Image</option>
                    </select>
                </div>
                <div class="form-group">
                    <label for="contentFile">Choose File</label>
                    <div class="file-upload">
                        <input type="file" id="contentFile" accept="video/*, image/*">
                        <label for="contentFile" class="file-upload-label">
                            <i class="fas fa-cloud-upload-alt"></i> Select File
                        </label>
                    </div>
                </div>
                <div class="form-group">
                    <label for="contentTitle">Title</label>
                    <input type="text" id="contentTitle" placeholder="Enter title...">
                </div>
                <div class="form-group">
                    <label for="contentSubject">Subject</label>
                    <select id="contentSubject">
                        <option value="math">Math</option>
                        <option value="science">Science</option>
                        <option value="history">History</option>
                        <option value="language">Language</option>
                        <option value="other">Other</option>
                    </select>
                </div>
                <div class="form-group">
                    <label for="contentDescription">Description</label>
                    <textarea id="contentDescription" placeholder="Describe your
content..."></textarea>
                </div>
                <div class="form-actions">
                    <button type="button" class="btn" id="cancelUpload">Cancel</button>
                    <button type="submit" class="btn btn-primary">Upload</button>
                </div>
            </form>
        </div>
    </div>

    <!-- Video Player Modal -->
    <div class="video-player" id="videoPlayer">
        <div class="player-container">
            <div class="player-header">
                <div class="player-title" id="playerTitle">Video Title</div>
                <button class="close-player" id="closePlayer"><i class="fas fa-times"></i></button>
            </div>
            <div class="player-video" id="playerVideo">
                <i class="fas fa-play-circle"></i> Video Player
            </div>
            <div class="player-info">
                <div class="player-stats">
```

```
                    <span id="playerViews">1.2K views</span>
                    <span id="playerDate">3 days ago</span>
                    <span id="playerSubject">Math</span>
                </div>
                <div class="player-actions">
                    <button class="player-action-btn" id="playerLikeBtn">
                        <i class="fas fa-thumbs-up"></i> <span id="playerLikes">234</span>
                    </button>
                    <button class="player-action-btn" id="playerShareBtn">
                        <i class="fas fa-share"></i> Share
                    </button>
                    <button class="player-action-btn" id="playerSaveBtn">
                        <i class="fas fa-bookmark"></i> Save
                    </button>
                    <button class="player-action-btn" id="playerFollowBtn">
                        <i class="fas fa-user-plus"></i> Follow
                    </button>
                    <button class="player-action-btn download-btn" id="playerDownloadBtn">
                        <i class="fas fa-download"></i> Download
                    </button>
                </div>
                <div class="player-description" id="playerDescription">
                    This is a description of the video content. It provides information about what the
video is about and what viewers can expect to learn.
                </div>
                <div class="comments-section">
                    <div class="comments-header">
                        <div class="comments-count">Comments (24)</div>
                        <select class="sort-comments">
                            <option>Top Comments</option>
                            <option>Newest First</option>
                            <option>Oldest First</option>
                        </select>
                    </div>
                    <div class="comment-form">
                        <input type="text" class="comment-input" placeholder="Add a comment...">
                        <button class="comment-submit">Post</button>
                    </div>
                    <div class="comments-container" id="commentsContainer">
                        <!-- Comments will be dynamically added here -->
                    </div>
                    <div class="load-more-comments">
                        <button class="load-more-btn" id="loadMoreComments">Load More
Comments</button>
```

```
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- Download Progress Modal -->
    <div class="download-modal" id="downloadModal">
        <div class="download-content">
            <h3 class="download-title">DOWNLOADING CONTENT</h3>
            <div class="download-progress">
                <div class="download-progress-bar" id="downloadProgressBar"></div>
                <div class="download-percentage" id="downloadPercentage">0%</div>
            </div>
            <div class="download-status" id="downloadStatus">Preparing download...</div>
        </div>
    </div>

    <!-- Toast Notification -->
    <div class="toast" id="toast">
        <div class="toast-icon"><i class="fas fa-check-circle"></i></div>
        <div class="toast-message" id="toastMessage">Action completed successfully!</div>
    </div>

    <footer>
        <div class="container">
            <p>&copy; 2023 StudyTube - Retro Educational Hub | All Rights Reserved</p>
        </div>
    </footer>

    <script>
        // Sample comments data
        const sampleComments = [
            { id: 1, author: "User123", initial: "U", text: "This video really helped me understand the concept better!", likes: 12, date: "2 hours ago" },
            { id: 2, author: "StudyBuddy", initial: "S", text: "Great explanation! Could you make a follow-up video on advanced topics?", likes: 8, date: "3 hours ago" },
            { id: 3, author: "MathNinja", initial: "M", text: "I've been struggling with this for weeks. Your teaching method is amazing!", likes: 25, date: "5 hours ago" },
            { id: 4, author: "PhysicsFan", initial: "P", text: "The visual aids really make a difference. Thank you for putting in the effort!", likes: 15, date: "6 hours ago" },
            { id: 5, author: "HistoryBuff", initial: "H", text: "This connects so well with what we learned in class yesterday.", likes: 7, date: "8 hours ago" },
      { id: 6, author: "ChemQueen", initial: "C", text: "Can you explain the molecular structure in more detail? Overall great content!", likes: 9, date: "10 hours ago" },
      { id: 7, author: "BioBrain", initial: "B", text: "This is exactly what I needed for my exam prep. Subscribed!", likes: 18, date: "12 hours ago" },
      { id: 8, author: "LangLearner", initial: "L", text: "The pronunciation guide at the end was super helpful. Thanks!", likes: 11, date: "1 day ago" },
      { id: 9, author: "CodeMaster", initial: "C", text: "I wish my professor explained it this clearly. Sharing with my study group!", likes: 22, date: "1 day ago" },
      { id: 10, author: "ArtStudent", initial: "A", text: "The historical context you provided really enhanced my understanding.", likes: 6, date: "1 day ago" },
      { id: 11, author: "EcoWarrior", initial: "E", text: "Love how you connected this to real-world applications. More videos like this please!", likes: 14, date: "2 days ago" },
      { id: 12, author: "MusicTheory", initial: "M", text: "This mathematical approach to music theory is fascinating!", likes: 19, date: "2 days ago" }
    ];

    // Sample data
    const sampleContent = [
      {
        id: 1,
        title: "Calculus Fundamentals Explained",
        author: "MathMaster",
        authorInitial: "M",
        views: 15420,
        likes: 892,
        date: "2 days ago",
        subject: "Math",
        duration: "15:42",
        thumbnail: "https://picsum.photos/seed/math1/400/225.jpg",
        type: "video",
        description: "This video covers the basics of calculus including limits, derivatives, and integrals.",
        downloadUrl: "https://example.com/download/calculus-fundamentals.mp4"
      },
      {
        id: 2,
        title: "Chemistry Lab Safety Guide",
        author: "ScienceWhiz",
        authorInitial: "S",
        views: 8930,
        likes: 421,
        date: "1 day ago",
        subject: "Science",
        duration: "8:30",
        thumbnail: "https://picsum.photos/seed/science1/400/225.jpg",
        type: "video",
        description: "Essential safety protocols for chemistry laboratory work.",
        downloadUrl: "https://example.com/download/chemistry-safety.mp4"
      },
      {
        id: 3,
        title: "World War II Timeline",
        author: "HistoryBuff",
        authorInitial: "H",
        views: 12350,
        likes: 567,
        date: "3 days ago",
        subject: "History",
        duration: "22:15",
        thumbnail: "https://picsum.photos/seed/history1/400/225.jpg",
        type: "video",
        description: "A comprehensive timeline of major events during World War II.",
        downloadUrl: "https://example.com/download/wwii-timeline.mp4"
      },
      {
        id: 4,
        title: "Spanish Grammar Cheat Sheet",
        author: "LanguageLover",
        authorInitial: "L",
        views: 6780,
        likes: 342,
        date: "1 day ago",
        subject: "Language",
        duration: "12:05",
        thumbnail: "https://picsum.photos/seed/language1/400/225.jpg",
        type: "video",
        description: "Quick reference guide for Spanish grammar rules and exceptions.",
        downloadUrl: "https://example.com/download/spanish-grammar.mp4"
      },
      {
        id: 5,
        title: "Physics Equations Explained",
        author: "PhysicsPro",
        authorInitial: "P",
        views: 9450,
        likes: 489,
        date: "2 days ago",
        subject: "Science",
                duration: "18:20",
                thumbnail: "https://picsum.photos/seed/physics1/400/225.jpg",
                type: "video",
                description: "Breaking down complex physics equations into understandable concepts.",
                downloadUrl: "https://example.com/download/physics-equations.mp4"
            },
            {
                id: 6,
                title: "Algebra Problem Solving Techniques",
                author: "MathMaster",
                authorInitial: "M",
                views: 11200,
                likes: 623,
                date: "3 days ago",
                subject: "Math",
                duration: "14:30",
                thumbnail: "https://picsum.photos/seed/math2/400/225.jpg",
                type: "video",
                description: "Step-by-step approaches to solving complex algebra problems.",
                downloadUrl: "https://example.com/download/algebra-techniques.mp4"
            },
            {
                id: 7,
                title: "Ancient Civilizations Infographic",
                author: "HistoryBuff",
                authorInitial: "H",
                views: 7650,
                likes: 387,
                date: "1 day ago",
                subject: "History",
                duration: "Image",
                thumbnail: "https://picsum.photos/seed/history2/400/225.jpg",
                type: "image",
                description: "Visual comparison of major ancient civilizations and their achievements.",
                downloadUrl: "https://example.com/download/ancient-civilizations.jpg"
            },
            {
                id: 8,
                title: "Biology Cell Structure Diagram",
                author: "ScienceWhiz",
                authorInitial: "S",
                views: 5420,
            likes: 298,
            date: "2 days ago",
            subject: "Science",
            duration: "Image",
            thumbnail: "https://picsum.photos/seed/biology1/400/225.jpg",
            type: "image",
            description: "Detailed diagram of plant and animal cell structures with labels.",
            downloadUrl: "https://example.com/download/cell-structure.jpg"
        }
];

// DOM Elements
const navTabs = document.querySelectorAll('.nav-tab');
const contentSections = document.querySelectorAll('.content-section');
const uploadBtn = document.getElementById('uploadBtn');
const uploadModal = document.getElementById('uploadModal');
const closeUploadModal = document.getElementById('closeUploadModal');
const cancelUpload = document.getElementById('cancelUpload');
const uploadForm = document.getElementById('uploadForm');
const videoPlayer = document.getElementById('videoPlayer');
const closePlayer = document.getElementById('closePlayer');
const exitBtn = document.getElementById('headerExitBtn');
const fixedExitBtn = document.getElementById('fixedExitBtn');
const toast = document.getElementById('toast');
const toastMessage = document.getElementById('toastMessage');
const searchBtn = document.getElementById('searchBtn');
const searchInput = document.getElementById('searchInput');
const contentFile = document.getElementById('contentFile');
const fileLabel = document.querySelector('.file-upload-label');
const commentsContainer = document.getElementById('commentsContainer');
const loadMoreComments = document.getElementById('loadMoreComments');
const downloadModal = document.getElementById('downloadModal');
const downloadProgressBar = document.getElementById('downloadProgressBar');
const downloadPercentage = document.getElementById('downloadPercentage');
const downloadStatus = document.getElementById('downloadStatus');

let currentComments = [];
let commentsLoaded = 0;
const commentsPerLoad = 5;
let currentPlayingItem = null;

// Initialize
document.addEventListener('DOMContentLoaded', () => {
    loadContent('trending', sampleContent);
```

```
            loadContent('subscriptions', sampleContent.filter(item => item.author === 'MathMaster' ||
item.author === 'ScienceWhiz'));
            loadContent('library', sampleContent.slice(0, 4));
            loadContent('my-videos', sampleContent.filter(item => item.author === 'MathMaster'));
            loadInitialComments();
        });

        // Load initial comments
        function loadInitialComments() {
            currentComments = [...sampleComments];
            commentsLoaded = 0;
            renderComments();
        }

        // Render comments
        function renderComments() {
            const commentsToShow = currentComments.slice(0, commentsLoaded +
commentsPerLoad);
            commentsContainer.innerHTML = '';

            commentsToShow.forEach(comment => {
                const commentElement = createCommentElement(comment);
                commentsContainer.appendChild(commentElement);
            });

            commentsLoaded += commentsPerLoad;

            // Hide load more button if all comments are loaded
            if (commentsLoaded >= currentComments.length) {
                loadMoreComments.style.display = 'none';
            } else {
                loadMoreComments.style.display = 'block';
            }
        }

        // Create comment element
        function createCommentElement(comment) {
            const commentDiv = document.createElement('div');
            commentDiv.className = 'comment';
            commentDiv.innerHTML = `
                <div class="comment-avatar">${comment.initial}</div>
                <div class="comment-content">
                    <div class="comment-author">${comment.author}</div>
                    <div class="comment-text">${comment.text}</div>
                    <div class="comment-actions">
                        <span class="comment-action like-action" data-id="${comment.id}">
                            <i class="fas fa-thumbs-up"></i> ${comment.likes}
                        </span>
                        <span class="comment-action">Reply</span>
                        <span class="comment-action">Share</span>
                        <span class="comment-action" style="color: rgba(51, 255, 51,
0.5);">${comment.date}</span>
                    </div>
                </div>
            `;

            // Add like functionality
            const likeAction = commentDiv.querySelector('.like-action');
            likeAction.addEventListener('click', function() {
                const currentLikes = parseInt(this.textContent.trim().split(' ')[1]);
                const isLiked = this.classList.contains('liked');

                if (isLiked) {
                    this.innerHTML = `<i class="fas fa-thumbs-up"></i> ${currentLikes - 1}`;
                    this.classList.remove('liked');
                } else {
                    this.innerHTML = `<i class="fas fa-thumbs-up"></i> ${currentLikes + 1}`;
                    this.classList.add('liked');
                    this.style.color = 'var(--danger-color)';
                }
            });

            return commentDiv;
        }

        // Load more comments
        loadMoreComments.addEventListener('click', () => {
            renderComments();
        });

        // Navigation
        navTabs.forEach(tab => {
            tab.addEventListener('click', () => {
                const tabName = tab.getAttribute('data-tab');

                // Update active tab
                navTabs.forEach(t => t.classList.remove('active'));
                tab.classList.add('active');
                // Show corresponding content
                contentSections.forEach(section => {
                    section.classList.remove('active');
                    if (section.id === tabName) {
                        section.classList.add('active');
                    }
                });
            });
        });

        // Load Content
        function loadContent(section, data) {
            const grid = document.getElementById(`${section}Grid`);
            grid.innerHTML = '';

            data.forEach(item => {
                const card = createContentCard(item);
                grid.appendChild(card);
            });
        }

        // Create Content Card
        function createContentCard(item) {
            const card = document.createElement('div');
            card.className = 'content-card';
            card.dataset.id = item.id;

            const durationDisplay = item.type === 'image' ? 'Image' : item.duration;

            card.innerHTML = `
                <div class="content-thumbnail">
                    <img src="${item.thumbnail}" alt="${item.title}">
                    <div class="content-duration">${durationDisplay}</div>
                </div>
                <div class="content-info">
                    <div class="content-title">${item.title}</div>
                    <div class="content-meta">
                        <div class="content-author">
                            <div class="author-avatar">${item.authorInitial}</div>
                            <span>${item.author}</span>
                        </div>
                        <div class="content-date">${item.date}</div>
                    </div>
```

```
                <div class="content-stats">
                    <span><i class="fas fa-eye"></i> ${formatNumber(item.views)}</span>
                    <span><i class="fas fa-thumbs-up"></i> ${formatNumber(item.likes)}</span>
                </div>
            </div>
            <div class="content-actions">
                <button class="action-btn like-btn" data-id="${item.id}">
                    <i class="fas fa-thumbs-up"></i> Like
                </button>
                <button class="action-btn comment-btn" data-id="${item.id}">
                    <i class="fas fa-comment"></i> Comment
                </button>
                <button class="action-btn share-btn" data-id="${item.id}">
                    <i class="fas fa-share"></i> Share
                </button>
                <button class="action-btn save-btn" data-id="${item.id}">
                    <i class="fas fa-bookmark"></i> Save
                </button>
                <button class="action-btn download-btn" data-id="${item.id}"
data-url="${item.downloadUrl}" data-title="${item.title}" data-type="${item.type}">
                    <i class="fas fa-download"></i> Download
                </button>
            </div>
        `;

        // Add event listeners
        card.querySelector('.content-thumbnail').addEventListener('click', () =>
openPlayer(item));

        const likeBtn = card.querySelector('.like-btn');
        likeBtn.addEventListener('click', (e) => {
            e.stopPropagation();
            toggleLike(likeBtn);
        });

        const commentBtn = card.querySelector('.comment-btn');
        commentBtn.addEventListener('click', (e) => {
            e.stopPropagation();
            openPlayer(item);
            // Focus on comment section
            setTimeout(() => {
                document.querySelector('.comment-input').focus();
            }, 300);
        });
```

```
            const shareBtn = card.querySelector('.share-btn');
            shareBtn.addEventListener('click', (e) => {
                e.stopPropagation();
                showToast('Link copied to clipboard!');
            });

            const saveBtn = card.querySelector('.save-btn');
            saveBtn.addEventListener('click', (e) => {
                e.stopPropagation();
                toggleSave(saveBtn);
            });

            const downloadBtn = card.querySelector('.download-btn');
            downloadBtn.addEventListener('click', (e) => {
                e.stopPropagation();
                downloadContent(item);
            });

            return card;
        }

        // Format Number
        function formatNumber(num) {
            if (num >= 1000000) {
                return (num / 1000000).toFixed(1) + 'M';
            } else if (num >= 1000) {
                return (num / 1000).toFixed(1) + 'K';
            }
            return num.toString();
        }

        // Open Player
        function openPlayer(item) {
            currentPlayingItem = item;
            document.getElementById('playerTitle').textContent = item.title;
            document.getElementById('playerViews').textContent = `${formatNumber(item.views)}
views`;
            document.getElementById('playerDate').textContent = item.date;
            document.getElementById('playerSubject').textContent = item.subject;
            document.getElementById('playerLikes').textContent = formatNumber(item.likes);
            document.getElementById('playerDescription').textContent = item.description;

            // Set player content based on type
            const playerVideo = document.getElementById('playerVideo');
            if (item.type === 'image') {
                playerVideo.innerHTML = `<img src="${item.thumbnail}" alt="${item.title}"
style="max-width: 100%; max-height: 100%;">`;
            } else {
                playerVideo.innerHTML = `<i class="fas fa-play-circle"></i> Video Player`;
            }

            // Reset comments
            loadInitialComments();

            videoPlayer.classList.add('active');
        }

        // Close Player
        closePlayer.addEventListener('click', () => {
            videoPlayer.classList.remove('active');
        });

        // Player Actions
        document.getElementById('playerLikeBtn').addEventListener('click', function() {
            toggleLike(this);
        });

        document.getElementById('playerShareBtn').addEventListener('click', () => {
            showToast('Link copied to clipboard!');
        });

        document.getElementById('playerSaveBtn').addEventListener('click', function() {
            toggleSave(this);
        });

        document.getElementById('playerFollowBtn').addEventListener('click', function() {
            const isFollowing = this.textContent.includes('Follow');
            if (isFollowing) {
                this.innerHTML = '<i class="fas fa-user-check"></i> Following';
                showToast('You are now following this creator!');
            } else {
                this.innerHTML = '<i class="fas fa-user-plus"></i> Follow';
                showToast('You have unfollowed this creator.');
            }
        });

        document.getElementById('playerDownloadBtn').addEventListener('click', () => {
            if (currentPlayingItem) {
                downloadContent(currentPlayingItem);
            }
        });

        // Download Content Function
        function downloadContent(item) {
            downloadModal.classList.add('active');
            downloadProgressBar.style.width = '0%';
            downloadPercentage.textContent = '0%';
            downloadStatus.textContent = 'Preparing download...';

            let progress = 0;
            const downloadInterval = setInterval(() => {
                progress += Math.random() * 15;
                if (progress >= 100) {
                    progress = 100;
                    clearInterval(downloadInterval);

                    downloadProgressBar.style.width = '100%';
                    downloadPercentage.textContent = '100%';
                    downloadStatus.textContent = 'Download complete!';

                    // Create download link
                    const link = document.createElement('a');
                    link.href = item.downloadUrl;
                    link.download = `${item.title.replace(/\s+/g, '_')}.${item.type === 'video' ? 'mp4' :
'jpg'}`;
                    link.style.display = 'none';
                    document.body.appendChild(link);

                    // Trigger download
                    setTimeout(() => {
                        link.click();
                        document.body.removeChild(link);

                        setTimeout(() => {
                            downloadModal.classList.remove('active');
                            showToast(`${item.title} downloaded successfully!`);
                        }, 500);
                    }, 500);
                } else {
                    downloadProgressBar.style.width = `${progress}%`;
                    downloadPercentage.textContent = `${Math.round(progress)}%`;
                    if (progress < 30) {
                        downloadStatus.textContent = 'Connecting to server...';
                    } else if (progress < 60) {
                        downloadStatus.textContent = 'Downloading content...';
                    } else if (progress < 90) {
                        downloadStatus.textContent = 'Finalizing download...';
                    } else {
                        downloadStatus.textContent = 'Almost done...';
                    }
                }
            }, 200);
        }

        // Toggle Like
        function toggleLike(btn) {
            btn.classList.toggle('liked');
            const isLiked = btn.classList.contains('liked');
            const icon = btn.querySelector('i');

            if (isLiked) {
                icon.classList.remove('fa-thumbs-up');
                icon.classList.add('fa-thumbs-up');
                showToast('You liked this content!');
            } else {
                showToast('You removed your like.');
            }
        }

        // Toggle Save
        function toggleSave(btn) {
            const isSaved = btn.style.color === 'var(--accent-color)';

            if (isSaved) {
                btn.style.color = '';
                showToast('Removed from library.');
            } else {
                btn.style.color = 'var(--accent-color)';
                showToast('Saved to library!');
            }
        }

        // Upload Modal
        uploadBtn.addEventListener('click', () => {
```

```
            uploadModal.classList.add('active');
        });

        closeUploadModal.addEventListener('click', () => {
            uploadModal.classList.remove('active');
        });

        cancelUpload.addEventListener('click', () => {
            uploadModal.classList.remove('active');
        });

        // File Upload
        contentFile.addEventListener('change', function() {
            const fileName = this.files[0]?.name || 'Select File';
            fileLabel.innerHTML = `<i class="fas fa-file"></i> ${fileName}`;
        });

        // Upload Form
        uploadForm.addEventListener('submit', (e) => {
            e.preventDefault();

            // Get form values
            const contentType = document.getElementById('contentType').value;
            const contentTitle = document.getElementById('contentTitle').value;
            const contentSubject = document.getElementById('contentSubject').value;
            const contentDescription = document.getElementById('contentDescription').value;

            // Create new content item
            const newItem = {
                id: sampleContent.length + 1,
                title: contentTitle,
                author: "CurrentUser",
                authorInitial: "Y",
                views: 0,
                likes: 0,
                date: "Just now",
                subject: contentSubject,
                duration: contentType === 'video' ? '0:00' : 'Image',
                thumbnail: `https://picsum.photos/seed/new${Date.now()}/400/225.jpg`,
                type: contentType,
                description: contentDescription,
                downloadUrl: `https://example.com/download/${contentTitle.replace(/\s+/g,
'_')}.${contentType === 'video' ? 'mp4' : 'jpg'}`
            };
```

```
    // Add to sample data
    sampleContent.unshift(newItem);

    // Add to My Videos section
    const myVideosGrid = document.getElementById('myVideosGrid');
    const newCard = createContentCard(newItem);
    myVideosGrid.insertBefore(newCard, myVideosGrid.firstChild);

    // Reset form and close modal
    uploadForm.reset();
    fileLabel.innerHTML = '<i class="fas fa-cloud-upload-alt"></i> Select File';
    uploadModal.classList.remove('active');

    // Show success message
    showToast('Content uploaded successfully!');

    // Switch to My Videos tab
    document.querySelector('[data-tab="my-videos"]').click();
});

// Exit Button Functionality
function handleExit() {
    const exitConfirmation = document.createElement('div');
    exitConfirmation.style.cssText = `
        position: fixed;
        top: 50%;
        left: 50%;
        transform: translate(-50%, -50%);
        background-color: var(--card-bg);
        border: 2px solid var(--danger-color);
        padding: 30px;
        z-index: 2000;
        text-align: center;
        box-shadow: 0 0 20px rgba(255, 51, 51, 0.5);
    `;

    exitConfirmation.innerHTML = `
        <h3 style="color: var(--danger-color); margin-bottom: 20px; font-size: 24px;">
            <i class="fas fa-exclamation-triangle"></i> EXIT CONFIRMATION
        </h3>
        <p style="margin-bottom: 20px; font-size: 20px;">
            Are you sure you want to exit StudyTube?<br>
            Your progress will be saved.
        </p>
        <div style="display: flex; gap: 15px; justify-content: center;">
            <button class="btn btn-danger" id="confirmExit">
                <i class="fas fa-sign-out-alt"></i> Yes, Exit
            </button>
            <button class="btn" id="cancelExit">
                <i class="fas fa-times"></i> Cancel
            </button>
        </div>
    `;

    document.body.appendChild(exitConfirmation);

    document.getElementById('confirmExit').addEventListener('click', () => {
        showToast('Thank you for using StudyTube! Goodbye!');
        setTimeout(() => {
            window.close();
            window.location.href = 'about:blank';
        }, 1500);
    });

    document.getElementById('cancelExit').addEventListener('click', () => {
        document.body.removeChild(exitConfirmation);
    });
}

// Exit button event listeners
exitBtn.addEventListener('click', handleExit);
fixedExitBtn.addEventListener('click', handleExit);

// Search Functionality
searchBtn.addEventListener('click', performSearch);
searchInput.addEventListener('keypress', (e) => {
    if (e.key === 'Enter') {
        performSearch();
    }
});

function performSearch() {
    const searchTerm = searchInput.value.toLowerCase().trim();

    if (!searchTerm) {
        showToast('Please enter a search term.');
        return;
    }

    // Filter content based on search term
    const filteredContent = sampleContent.filter(item =>
        item.title.toLowerCase().includes(searchTerm) ||
        item.subject.toLowerCase().includes(searchTerm) ||
        item.author.toLowerCase().includes(searchTerm) ||
        item.description.toLowerCase().includes(searchTerm)
    );

    // Load filtered content in the trending section
    loadContent('trending', filteredContent);

    // Switch to trending tab
    document.querySelector('[data-tab="trending"]').click();

    // Show results message
    showToast(`Found ${filteredContent.length} results for "${searchTerm}"`);
}

// Show Toast Notification
function showToast(message) {
    toastMessage.textContent = message;
    toast.classList.add('show');

    setTimeout(() => {
        toast.classList.remove('show');
    }, 3000);
}

// Notification Button
document.getElementById('notificationBtn').addEventListener('click', () => {
    showToast('You have 3 new notifications!');
});

// Profile Button
document.getElementById('profileBtn').addEventListener('click', () => {
    showToast('Profile page coming soon!');
});

// Filter Buttons
document.querySelectorAll('.filter-btn').forEach(btn => {
    btn.addEventListener('click', function() {
        // Get current active section
```

```
                const activeSection = document.querySelector('.content-section.active').id;

                // Get filter text
                const filter = this.textContent;

                // Filter content based on the selected filter
                let filteredContent;

                if (filter === 'All Subjects' || filter === 'All') {
                    filteredContent = sampleContent;
                } else if (filter === 'Math' || filter === 'Science' || filter === 'History' || filter ===
'Languages') {
                    filteredContent = sampleContent.filter(item =>
                        item.subject.toLowerCase() === filter.toLowerCase() ||
                        (filter === 'Languages' && item.subject === 'Language')
                    );
                } else if (filter === 'Videos') {
                    filteredContent = sampleContent.filter(item => item.type === 'video');
                } else if (filter === 'Images') {
                    filteredContent = sampleContent.filter(item => item.type === 'image');
                } else {
                    // For time-based filters, just use all content for demo
                    filteredContent = sampleContent;
                }

                // Load filtered content
                loadContent(activeSection, filteredContent);

                // Update active filter button style
                document.querySelectorAll('.filter-btn').forEach(b => b.style.backgroundColor = '');
                this.style.backgroundColor = 'rgba(51, 255, 51, 0.2)';
            });
        });

        // Comment Submit
        document.querySelector('.comment-submit').addEventListener('click', () => {
            const commentInput = document.querySelector('.comment-input');
            const commentText = commentInput.value.trim();

            if (!commentText) {
                showToast('Please enter a comment.');
                return;
            }
            // Create new comment
            const newComment = {
                id: currentComments.length + 1,
                author: "You",
                initial: "Y",
                text: commentText,
                likes: 0,
                date: "Just now"
            };

            // Add to beginning of comments array
            currentComments.unshift(newComment);

            // Re-render comments
            loadInitialComments();

            // Clear input
            commentInput.value = '';

            // Show success message
            showToast('Comment posted successfully!');

            // Update comment count
            document.querySelector('.comments-count').textContent = `Comments
(${currentComments.length})`;
        });

        // Sort comments
        document.querySelector('.sort-comments').addEventListener('change', function() {
            const sortBy = this.value;

            if (sortBy === 'Newest First') {
                currentComments.sort((a, b) => new Date(b.date) - new Date(a.date));
            } else if (sortBy === 'Oldest First') {
                currentComments.sort((a, b) => new Date(a.date) - new Date(b.date));
            } else {
                // Top Comments (by likes)
                currentComments.sort((a, b) => b.likes - a.likes);
            }

            loadInitialComments();
        });
    </script>
</body>
```

```
</html>
```