

**TRƯỜNG CAO ĐẲNG THỰC HÀNH  
FPT-POLYTECHNIC TP.HCM**



**BÁO CÁO ASSIGNMENT**

- ✓ Mã số: PS24414
- ✓ Họ và tên: Trần Anh Vũ
- ✓ Lớp: CP17309\_3
- ✓ GVHD: Cô Mai Thanh Thảo

TP. HỒ CHÍ MINH - 2022

## I. Ý Tưởng:

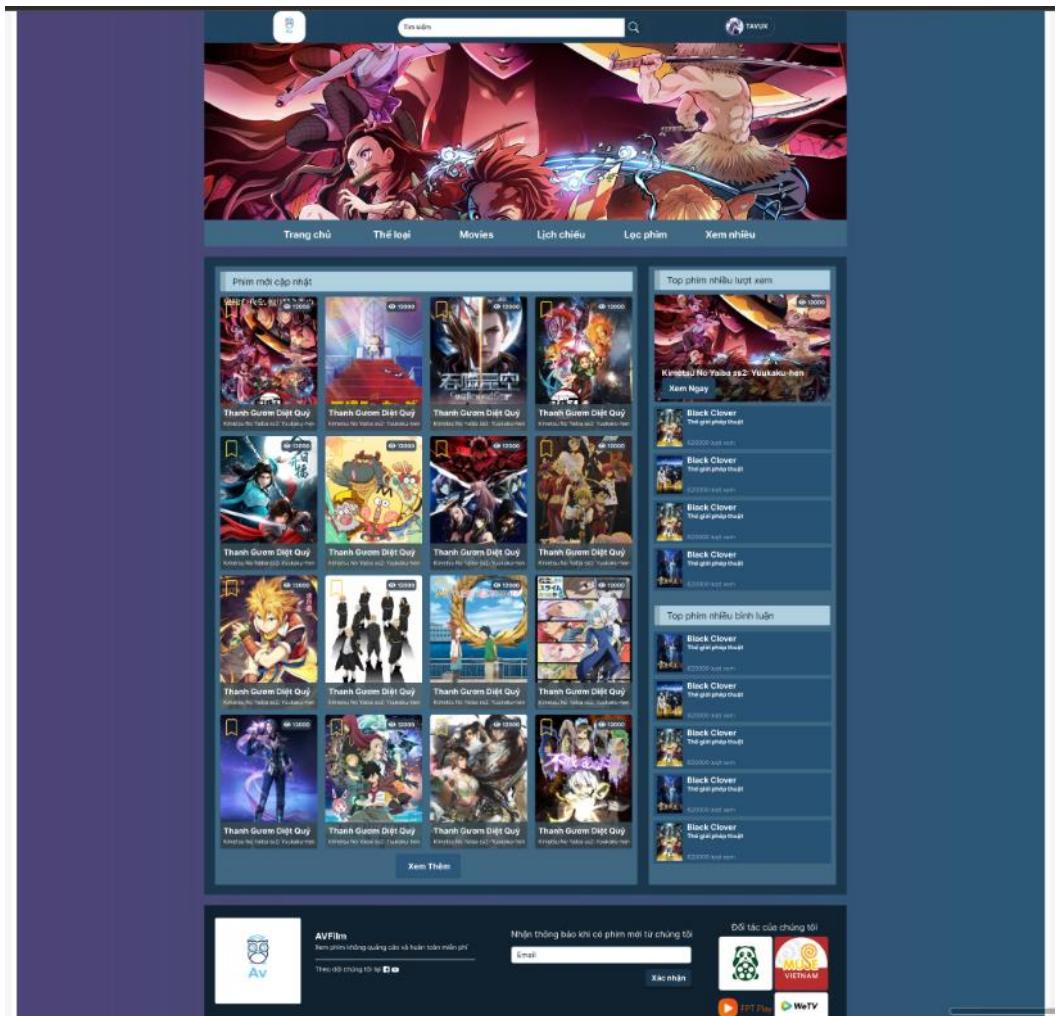
Website được sử dụng để xem phim hoạt hình với nhiều thể loại và hoàn toàn miễn phí.

Website sẽ gồm 6 trang con

- Trang chủ
- Trang chi tiết phim
- Đăng ký, Đăng nhập
- Trang quản lý tài khoản
- Trang thể loại
- Trang lịch chiếu
- Trang xem phim

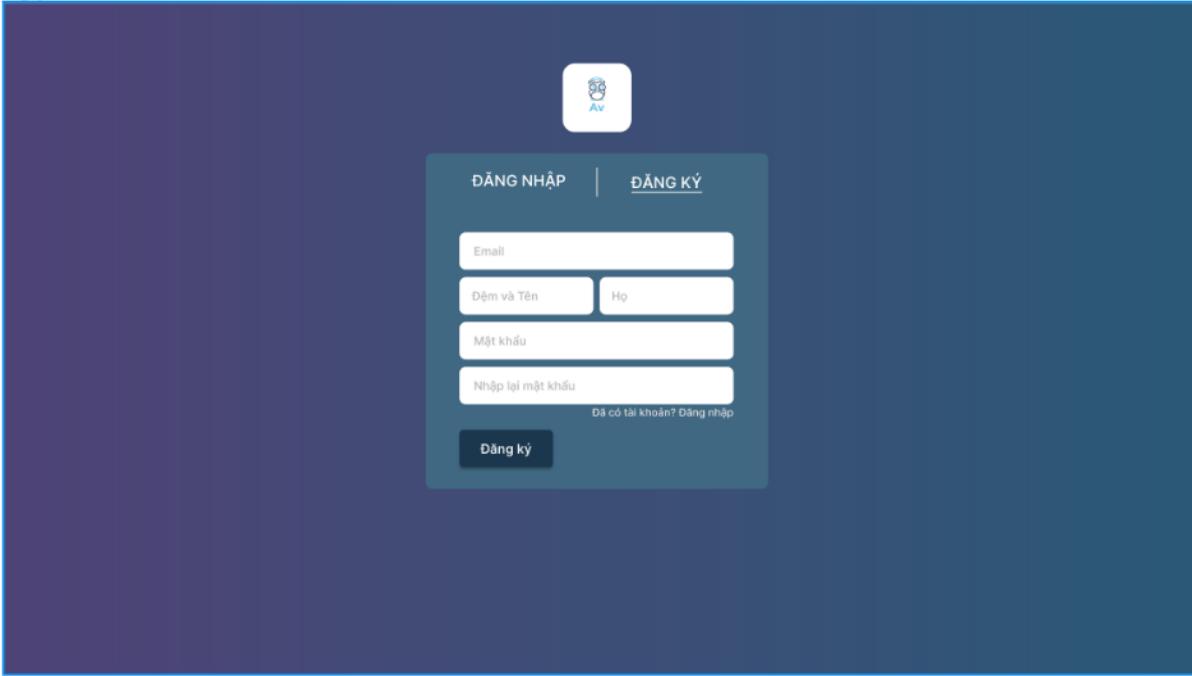
## II. Layout Website:

- Trang Chủ



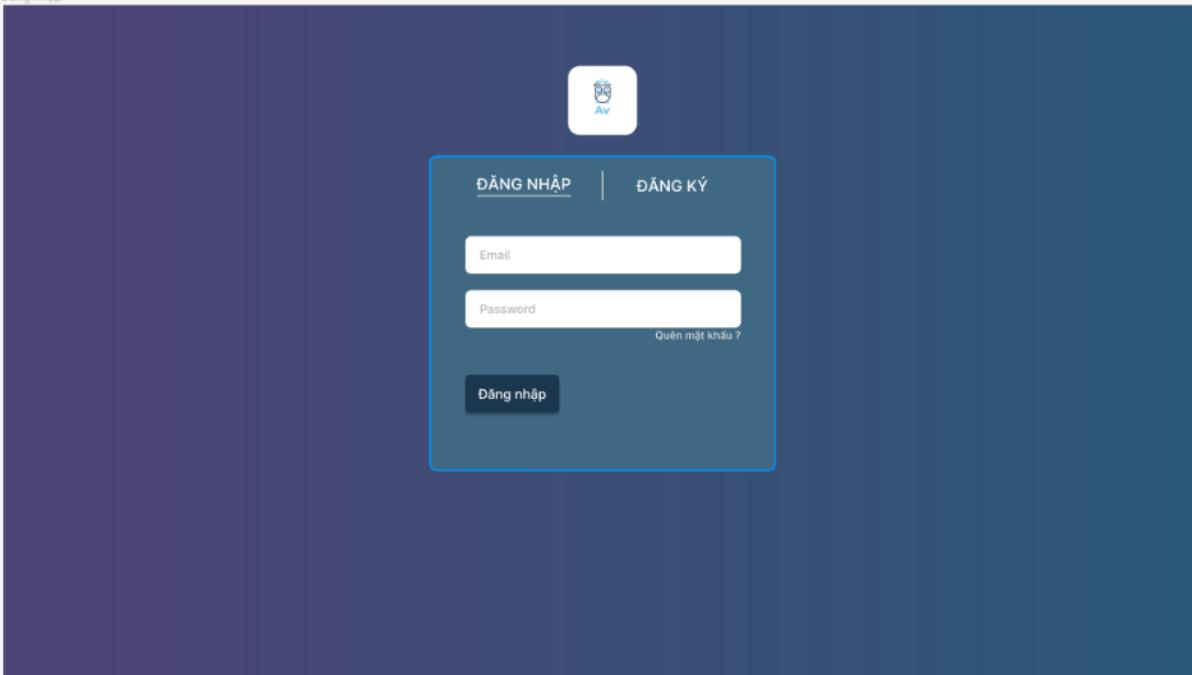
- Trang đăng ký, đăng nhập

Dăng ký



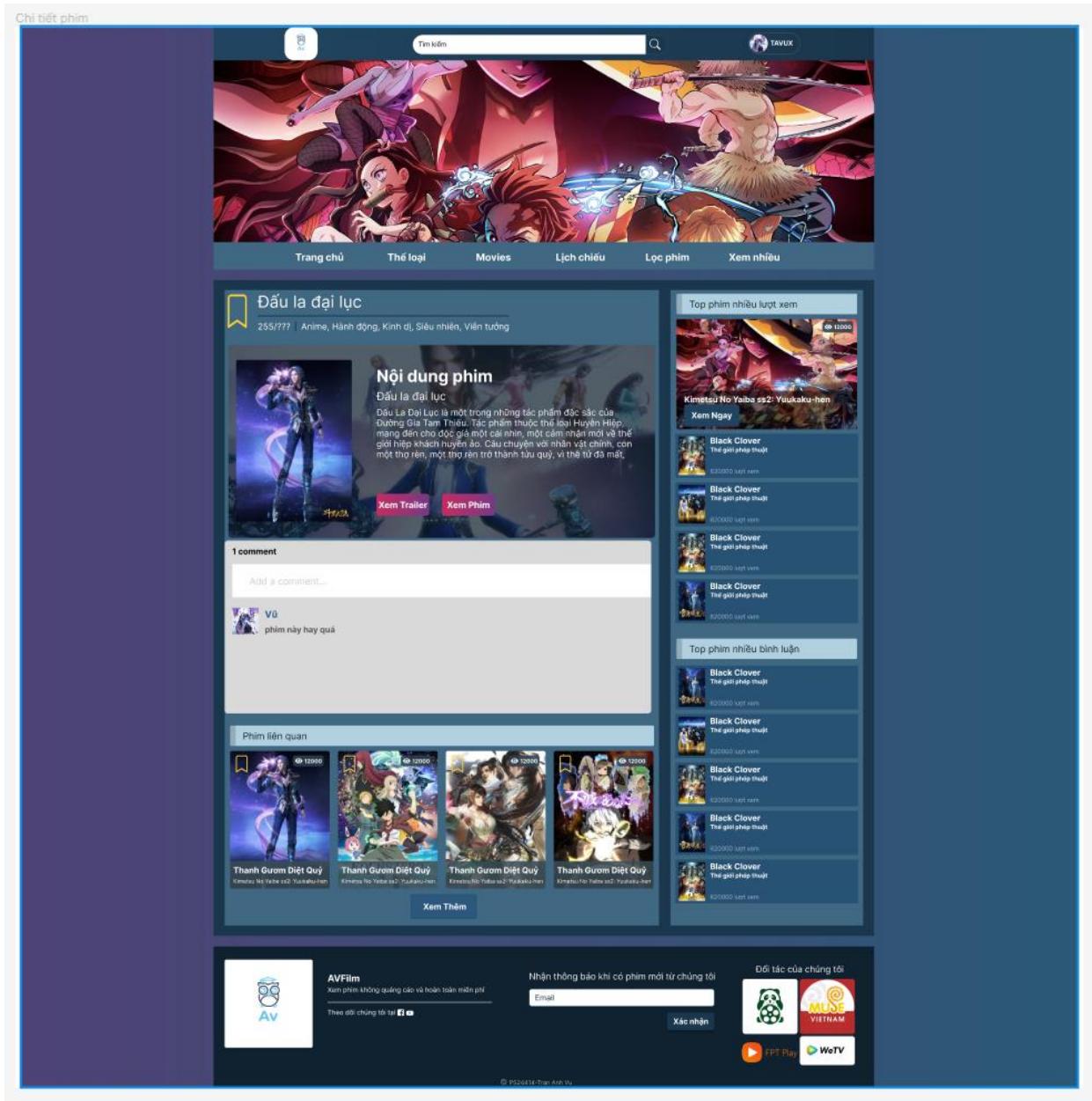
The registration form interface features a dark blue header with the text "ĐĂNG KÝ" underlined. Below the header is a white input field labeled "Email". To its right are two smaller input fields: "Đệm và Tên" and "Họ". Further down are fields for "Mật khẩu" and "Nhập lại mật khẩu". At the bottom left is a link "Đã có tài khoản? Đăng nhập" and a large dark blue "Đăng ký" button.

Dăng nhập



The login form interface features a dark blue header with the text "ĐĂNG NHẬP" underlined. Below the header are two input fields: "Email" and "Password". To the right of the "Password" field is a link "Quên mật khẩu ?". At the bottom left is a link "Đã có tài khoản? Đăng ký" and a large dark blue "Đăng nhập" button.

- Trang chi tiết phim



**Đấu La Đại Lục**

255/??? | Anime, Hành động, Kinh dị, Siêu nhiên, Viễn tưởng

**Nội dung phim**

Đấu La Đại Lục là một trong những tác phẩm đặc sắc của Đường Giả Tam Thiếu. Tác phẩm thuộc thể loại Huyền Huyễn, mang đến cho độc giả một cái nhìn, một cảm nhận mới về thế giới hiệp khách huyền ảo. Câu chuyện với nhân vật chính, con một thư rěn, một thử rěn trở thành tinh quí, vì thế từ đó mờ...

Xem Trailer Xem Phim

1 comment

Add a comment...

Võ phim này hay quá

**Phim liên quan**

Thanh Gươm Diệt Quỷ (Kimetsu No Yaiba ss2: Yuukaku-hen) Thanh Gươm Diệt Quỷ (Kimetsu No Yaiba ss2: Yuukaku-hen) Thanh Gươm Diệt Quỷ (Kimetsu No Yaiba ss2: Yuukaku-hen) Thanh Gươm Diệt Quỷ (Kimetsu No Yaiba ss2: Yuukaku-hen)

Xem Thêm

**Top phim nhiều lượt xem**

- Kimetsu No Yaiba ss2: Yuukaku-hen Xem Ngay
- Black Clover Thế giới phép thuật

**Top phim nhiều bình luận**

- Black Clover Thế giới phép thuật

**Đối tác của chúng tôi**

AVFilm Xem phim không quảng cáo và hoàn toàn miễn phí

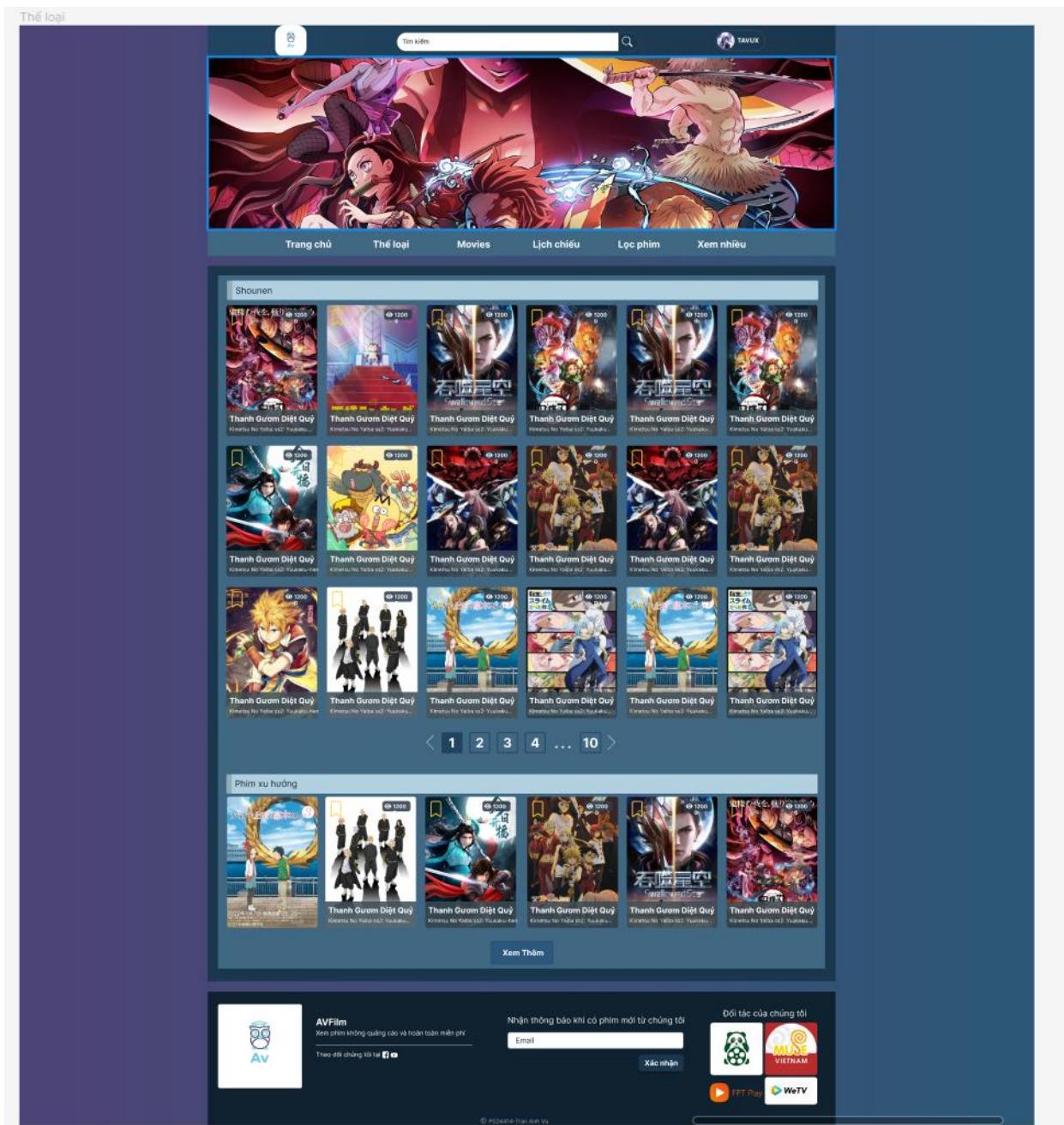
Theo dõi chúng tôi tại: [Facebook](#) [Twitter](#)

Nhận thông báo khi có phim mới từ chúng tôi

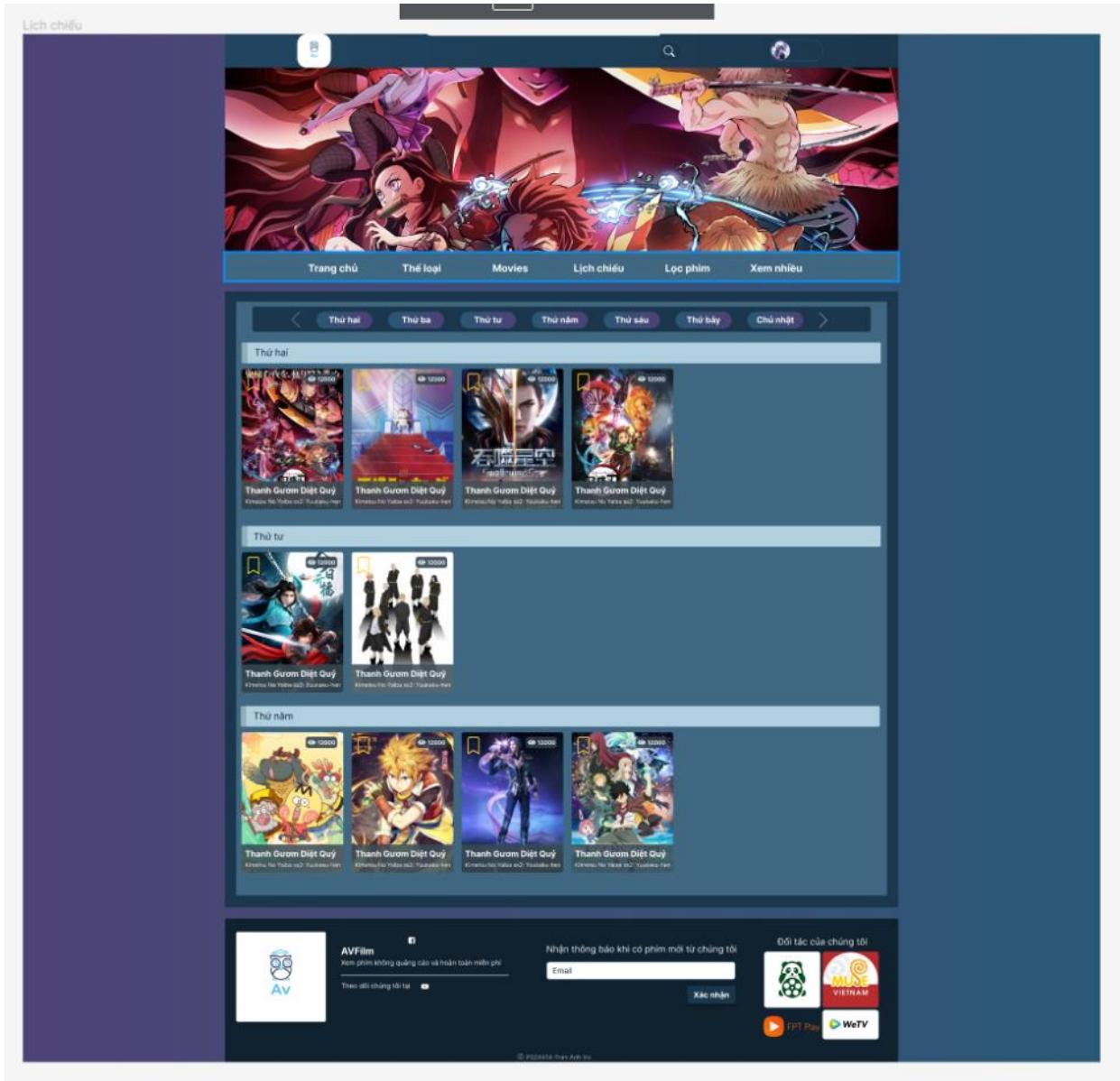
Email:  Xác nhận:

FPT Play WeTV

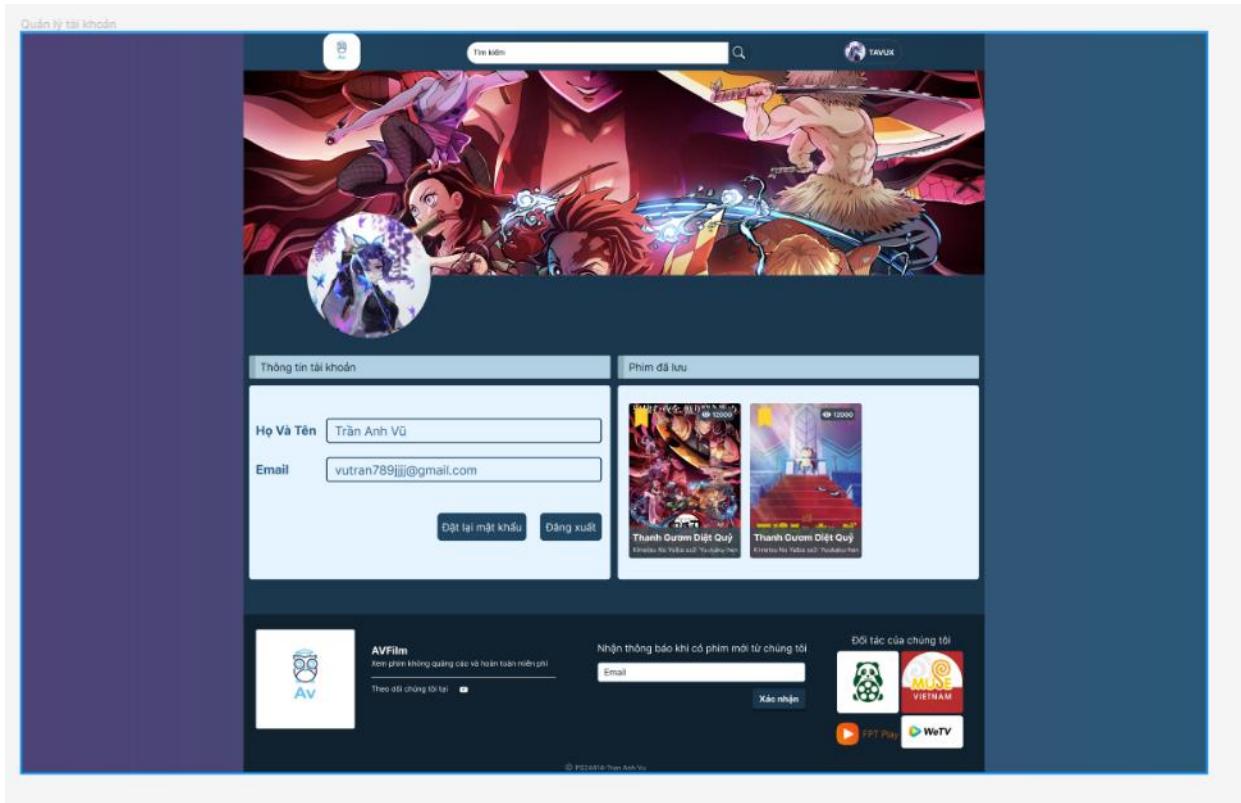
- Trang danh mục phim



- Trang lịch chiếu



- Trang quản lý tài khoản

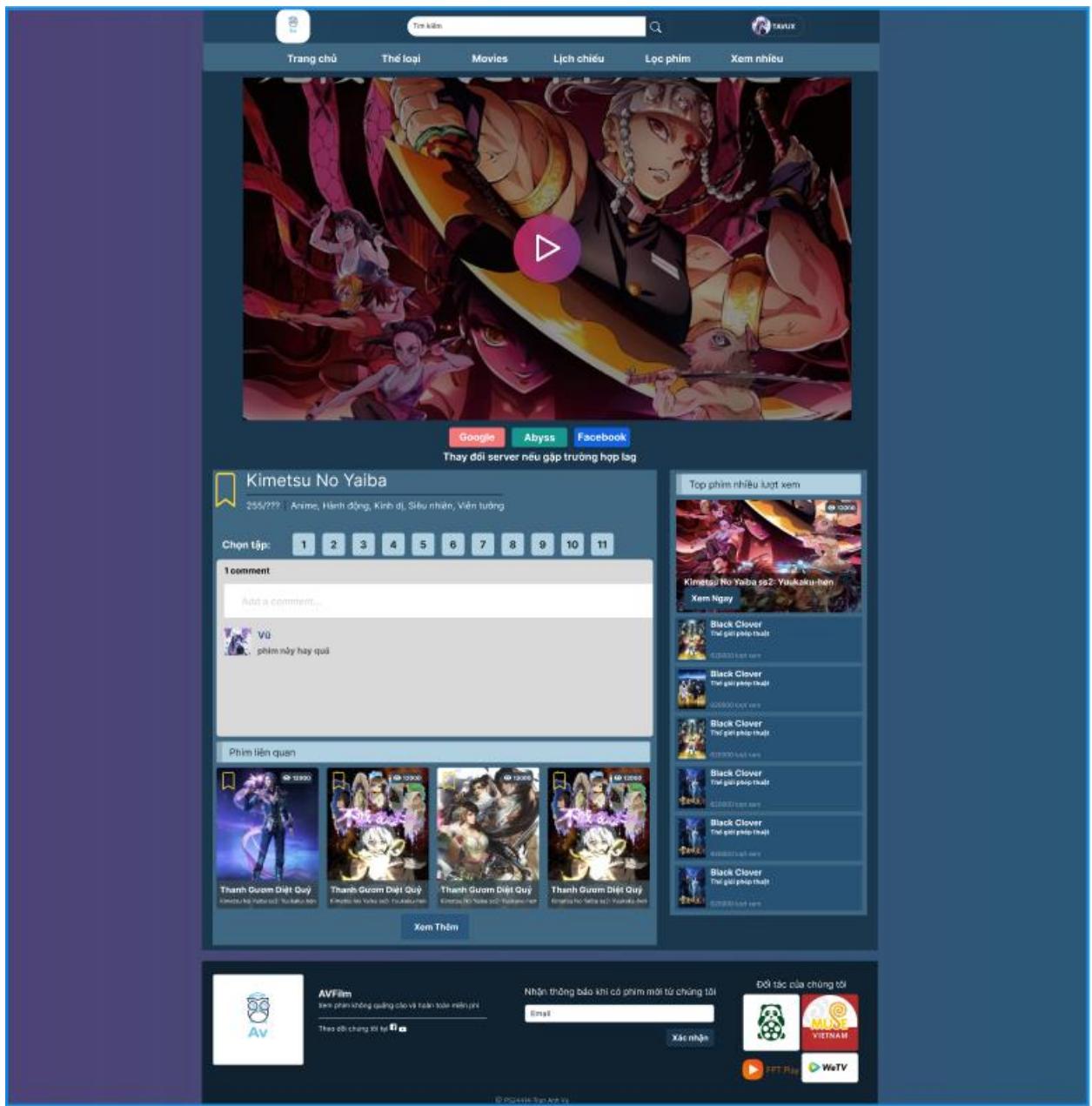


The screenshot shows the FPT Polytechnic account management interface. At the top, there is a search bar and a user profile icon labeled 'TAVUX'. Below the header, there is a large, vibrant anime-style illustration of several characters in dynamic poses.

The main content area is divided into sections:

- Thông tin tài khoản:** Displays the user's name ('Trần Anh Vũ') and email ('vutran789iii@gmail.com'). It includes buttons for 'Đặt lại mật khẩu' (Change password) and 'Đăng xuất' (Logout).
- Phim đã lưu:** Shows two movie thumbnails: 'Thanh Gươm Diệt Quỷ' (Kamen Rider Kabuto: Yuusha no Hito) and 'Thanh Gươm Diệt Quỷ' (Kamen Rider Kabuto: Yuusha no Hito).
- AVFilm:** A section for watching movies. It features a logo with a stylized eye, the text 'Xem phim không quảng cáo và hoàn toàn miễn phí' (Watch movies without ads and completely free), and a button 'Theo dõi chúng tôi' (Follow us). It also includes fields for 'Nhận thông báo khi có phim mới từ chúng tôi' (Receive notifications when new movies are released) and 'Email'.
- Đối tác của chúng tôi:** A section featuring logos for 'FPT Play', 'WeTV', and 'Muse Vietnam'.

- Trang xem phim



[Link Figma của bài thiết kế](#)

### III. Giải thích code:

- Trang chủ

```
<body>
  <div class="web-container">...
    </div>
```

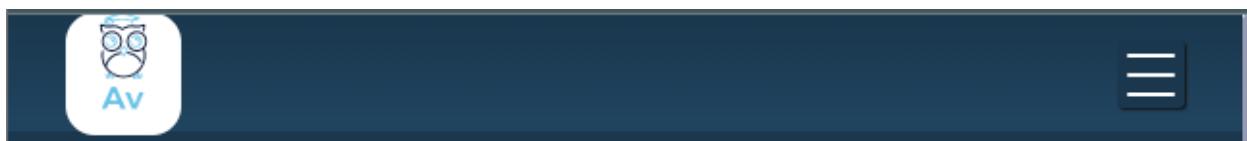
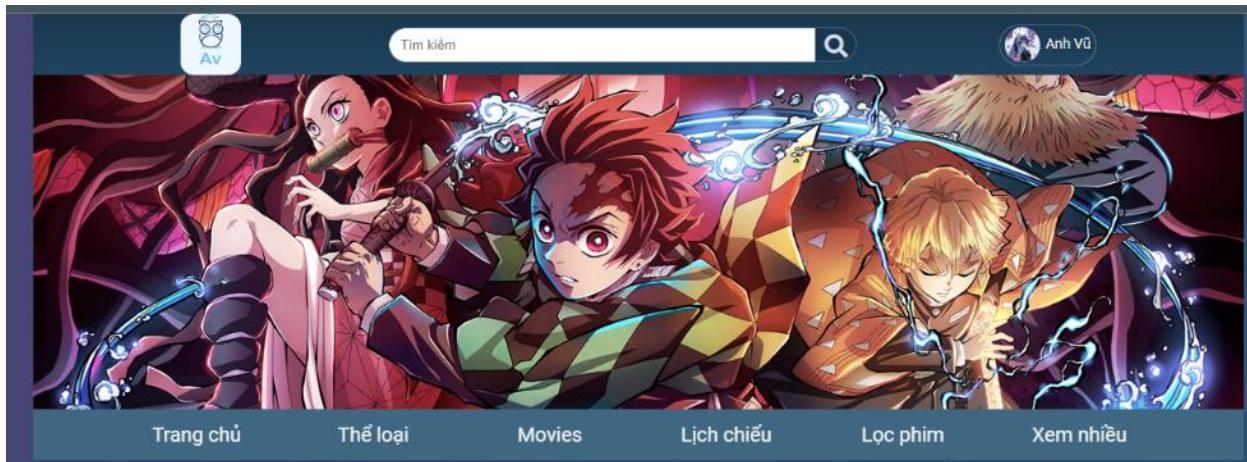
Toàn bộ website sẽ được bọc bởi class “web-container”

Nó sẽ có chiều rộng là 1200px. Max-width: 100% là giành cho trường hợp nếu trình duyệt có kích thước bé hơn 1200px thì “web-container” sẽ có kích thước ứng với kích thước của trình duyệt. Margin: 0 auto; là để căn giữa nó so với thẻ cha.

```
html {
  box-sizing: border-box;
  font-family: 'Roboto', sans-serif;
  font-size: 16px;
  line-height: 1.5px;
  background: #2b5876;
  background: -webkit-linear-gradient(to right, #4e4376, #2b5876);
  background: linear-gradient(to right, #4e4376, #2b5876);
}

.web-container {
  width: 1200px;
  max-width: 100%;
  margin: 0 auto;
}
```

## Header:



The main content area displays a grid of movie posters under the heading 'Phim mới cập nhật' (New movies updated). The visible titles are 'Tamako Market' (with 12,000 views) and 'Tamako Love Story' (with 30,000 views). To the right is a sidebar with a search bar and a list of links: Trang chủ (Home), Thể loại (Genre), Movies, Lịch chiếu (Release schedule), Lọc phim (Filter), and Xem nhiều (Most viewed). At the bottom of the sidebar is a user profile section for 'Anh Vũ'.

Header bao gồm header mobile (lúc màn hình có kích thước của điện thoại, tablet) và header mặc định.

Header bao gồm 3 phần nhỏ là: Header top, Header center, Header bottom.

Header top:

```

<!-- Header mobile -->
<header class="header-mobile">...
</header>

<!-- Header -->
<header class="header-top">
    <a href="#" class="logo">
        
    </a>
    <div class="searchbar-container">
        <input type="text" class="searchbar" placeholder="Tìm kiếm">
        <i class="fas fa-search search-icon"></i>
    </div>

    <!-- Login -->
    <!-- <div class="usercontainer">
        
        <span class="user-name">Vũ</span>
        <ul class="user-list">
            <li class="list-item">
                <a href="#"><i class="fas fa-user item-icon"></i></a>
                <a href="#">Quản lý tài khoản</a>
            </li>
            <li class="list-item">
                <a href="#"><i class="fas fa-bookmark item-icon"></i></a>
                <a href="#">Phim đã lưu</a>
            </li>
            <li class="list-item">
                <a href="#"><i class="fas fa-sign-out-alt item-icon"></i></a>
                <a href="#">Đăng xuất</a>
            </li>
        </ul>
    </div> -->

    <!-- None Login -->
    <div class="usercontainer ">
        <!-- <span class="user-name none-login">Tài khoản</span>
        <ul class="user-list list-none-login">
            ...
        </ul>
    </div>
</header>

<!-- None Login -->
<div class="usercontainer ">
    <!-- <span class="user-name none-login">Tài khoản</span>
    <ul class="user-list list-none-login">
        <li class="list-item">
            <a href="login.html"><i class="fas fa-sign-in-alt item-icon"></i></a>
            <a href="login.html">Đăng nhập</a>
        </li>
        <li class="list-item">
            <a href="register.html"><i class="fas fa-user-plus item-icon"></i></a>
            <a href="register.html">Đăng ký</a>
        </li>
    </ul> -->
</div>
</header>
```

Nó sẽ có chiều rộng vừa khớp với thẻ cha chứa nó và chiều rộng là 60px. Position: sticky; và top: 0px là giúp cho header-top luôn nằm ở trên đầu của trang. Layout trong header-top được sử dụng flex để giàn với thuộc tính justify-content:

space-evenly; để các item có thể cách đều nhau kể cả đối với méo của box. Align-item: center giúp cho các item có thể nằm giữa theo chiều dọc đối với header-top.

Header-center:

```
.header-center {  
    width: 100%;  
    height: 330px;  
    position: relative;  
    top: 0;  
    background-image: url(..../images/background-image.jpg);  
    background-size: cover;  
    background-position: bottom;  
}
```

```
</header>  
<header class="header-center">  
    <a href="#">  
        <div class="header-center-film">  
            <p class="film-name">Kimetsu No Yaiba ss2: Yuukaku-hen</p>  
            <p class="film-name1">Thanh Gươm Diệt Quỷ Phần 2 Phố Đèn Đỏ</p>  
            <button class="watch-btn btn">Xem ngay</button>  
        </div>  
    </a>  
</header>
```

Có chiều rộng khớp với thẻ cha và chiều cao là 330px. Position: relative; giúp cho việc căn layout cho con của nó. Background-size: cover; giúp cho ảnh nền có thể tự căn chỉnh để vừa với kích thước mà không bị bể ảnh. Background-position: bottom; sẽ giúp chuyển phần hiện thị của ảnh xuống phía dưới.

## Header-bottom:

```
<header class="header-bottom">
  <ul class="navbar">
    <li class="nav-item">
      <a href="index.html" class="nav-link">Trang chủ</a>
    </li>
    <li class="nav-item">
      <a href="#" class="nav-link">Thể loại</a>
      <ul class="menu-child">
        <li class="child-item">
          <a href="category.html" class="item-category">Chuyển sinh</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Shounen</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Shoujo</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Comedy</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Hài hước</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Thám tử</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Phiêu lưu</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Xuyên không</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Siêu năng lực</a>
        </li>
        <li class="child-item">
          <a href="category.html" class="item-category">Âm nhạc</a>
        </li>
        <li class="child-item">

```

```
.header-bottom {
  width: 100%;
  height: 50px;
  position: sticky;
  position: -webkit-sticky;
  top: 59px;
  z-index: 4;
  margin-bottom: 10px;
  background-color: #406882;
  box-shadow: 1px 1px 1px #244b69;
}
```

Nó có chiều rộng khớp với chiều rộng của thẻ cha chứa nó và chiều cao là 50px. Position: sticky; top: 59px; giúp nó có thể giữ nguyên vị trí khi còn cách phía trên của trang web 59px. Z-index: 4; giúp nó có thể nổi lên khi gấp các thẻ có

thuộc tính position khác. Ngoài ra các thuộc tính như box-shadow, background-color giúp đổ bóng cũng như đổ màu nền cho nó.

```
@media only screen and (max-width: 46.25em) {  
    .section {  
        flex-wrap: wrap;  
    }  
  
    .aside {  
        width: 100%;  
    }  
  
    /* ẩn header */  
    .header-top, .header-bottom, .header-center {  
        display: none;  
    }  
  
    /* ----- */
```

Khi trình duyệt có kích thước bé hơn hoặc bằng 46.25em => 740px thì sẽ ẩn header-top header-center và header-bottom sau đó thay thế bằng header mobile.

```


<header class="header-mobile">
    <div class="header-logo-mobile">
        <a href="index.html"></a>
    </div>
    <div class="header-nav-btn">
        <div class="roll"></div>
        <div class="roll"></div>
        <div class="roll"></div>
    </div>
    <div class="nav-menu_mobile">
        <div class="menu-search-item">
            <div class="search-bar-mobile">
                <input type="text" class="search-bar-mobile" placeholder="Tìm kiếm">
                <i class="fas fa-search"></i>
            </div>
        </div>
        <div class="menu_mobile-item"><a href="index.html" class="link_menu">Trang chủ</a></div>
        <div class="menu_mobile-item">
            <a href="#" class="link_menu" id="btnCategory">Thể loại</a>
            <div class="category-mobile">
                <div class="mobile-category-container">
                    <li class="child-item child-item-mobile">
                        <a href="category.html" class="item-category">Chuyển sinh</a>
                    </li>
                    <li class="child-item child-item-mobile">
                        <a href="category.html" class="item-category">Shounen</a>
                    </li>
                    <li class="child-item child-item-mobile">
                        <a href="category.html" class="item-category">Shoujo</a>
                    </li>
                    <li class="child-item child-item-mobile">
                        <a href="category.html" class="item-category">Comedy</a>
                    </li>
                    <li class="child-item child-item-mobile">
                        <a href="category.html" class="item-category">Hài hước</a>
                    </li>
                    <li class="child-item child-item-mobile">
                        <a href="category.html" class="item-category">Thám tử</a>
                    </li>
                </div>
            </div>
        </div>
        <div class="menu_mobile-item"><a href="index.html" class="link_menu">Trang chủ</a></div>
        <div class="menu_mobile-item">...
    </div>
    <div class="menu_mobile-item"><a href="#" class="link_menu">Movies</a></div>
    <div class="menu_mobile-item"><a href="Showtimes.html" class="link_menu">Lịch chiếu</a></div>
    <div class="menu_mobile-item"><a href="#" class="link_menu">Lọc phim</a></div>
    <div class="menu_mobile-item"><a href="#" class="link_menu">Xem nhiều</a></div>
    <div class="user">
        <div class="user-mobile-container" id="useMobileContainer">
            <!-- 
            <div class="user-name-mobile">TAVUX</div> -->
        </div>
        <i class="fas fa-sign-out-alt mobile-logout" id="bntLogoutMobile"></i>
    </div>
    <div class="gb-layer"></div>
</header>

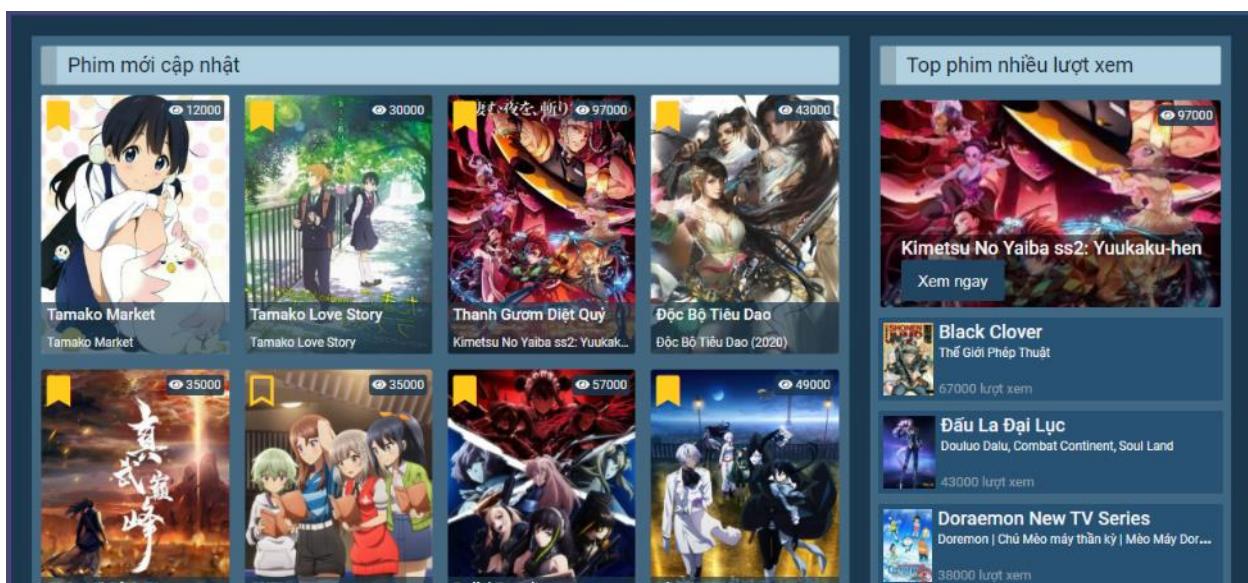
```

```
.header-mobile {
    width: 100%;
    height: 60px;
    position: sticky;
    position: -webkit-sticky;
    top: 0;
    display: flex;
    justify-content: space-between;
    padding: 0 30px;
    align-items: center;
    background: linear-gradient(to top, #214561, #1A374D);
    z-index: 7;
    /* flex-wrap: wrap; */
}
```

Header-mobile sẽ có chiều dài ứng với chiều dài của thẻ cha và chiều rộng là 60px. Position: sticky; top: 0; giúp nó có thể giữ nguyên vị trí khi còn cách phía trên của trang web 0. Display: flex; giúp dàn layout, justify-content: space-between; giúp cho các item sẽ đứng sang 2 bên của thẻ và chừa ra khoảng trống chính giữa. Align-item: center giúp căn giữa theo chiều dọc cho các item so với nó.

```
nv > scripts > 33 main.js ...
1 // Nút mở menu
2 var nav_btn = document.querySelector('.header-nav-btn');
3 // Menu
4 var nav_menu = document.querySelector(".nav-menu_mobile");
5 // Lớp phủ phía sau menu được kích hoạt sau khi bấm vào nav_btn
6 var bg_layer = document.querySelector(".gb-layer");
7 // nút the loại
8 var btn_category = document.getElementById("btnCategory");
9 // thẻ chứa các thể loại phim
10 var categoryListContainer = document.querySelector(".category-mobile");
11 // có chức năng bật menu khi bấm vào và tắt khi bấm vào 1 lần nữa
12 function turnOnOffMenuMobile() {
13     if (nav_btn.classList == "header-nav-btn exit") {
14         categoryListContainer.classList = "category-mobile";
15         nav_btn.classList = "header-nav-btn active";
16     } else {
17         nav_btn.classList.toggle("active");
18         nav_menu.classList.toggle("show");
19         bg_layer.classList.toggle("show")
20     }
21 }
22 nav_btn.onclick = function () {
23     turnOnOffMenuMobile();
24 }
25 bg_layer.onclick = function () {
26     turnOnOffMenuMobile();
27 }
28 // sẽ bắt ra tab thẻ loại phim khi bấm vào và tắt khi bấm vào lần nữa
29 btn_category.addEventListener("click", () => {
30     nav_btn.classList = "header-nav-btn exit";
31     categoryListContainer.classList = "category-mobile show";
32 })
```

Javascripts sử dụng trong header-mobile.



```
<!-- Section -->
<main class="section">
    <div class="article">
        <div class="article-container">
            <div class="film-category-header">Phim mới cập nhật</div>
            <div class="film-category-container" id="filmCategoryContainer"></div>
            <a href="#" class="other-link">
                <button class="other-btn btn">Xem thêm</button>
            </a>
        </div>
    </div>

    <div class="aside" id="aside"></div>
</main>
```

```
.section {
    width: 100%;
    min-height: 580px;
    background-color: #1A374D;
    display: flex;
    justify-content: space-evenly;
    padding-bottom: 20px;
}
```

Section là thẻ cha chứa các thẻ article (sẽ chứa những bộ phim trên website) và aside (sẽ chứa những phim nằm trên top lượt xem cũng như lượt bình luận). Nó sẽ có chiều rộng khớp với chiều rộng của thẻ chứa nó và chiều cao tối thiểu là

580px. Display: flex; được dùng để cho 2 thẻ article và aside nằm lên 1 hàng và justify-content: space-evenly; giúp cho các thẻ sẽ cách đều nhau. Padding-bottom: 20px; giúp cho thẻ sẽ có 1 khoảng trống tầm 20px

```
.article {
    width: 65.83%;
    margin-top: 20px;
    height: max-content;
    background-color: #406882;
}
```

Article là thẻ chứa phim trên website có chiều rộng là 65.83% so với thẻ cha của nó và cách phí trên nó 20px. Chiều cao vừa với kích thước lớn nhất của con trong nó.

```
.aside {
    width: 29.16%;
    height: fit-content;
    padding-bottom: 10px;
    margin-top: 20px;
    background-color: #406882;
    overflow: hidden;
}
```

Aside chứa các phim nằm trong top bình luận và top lượt xem. Có chiều rộng là 29.16% và chiều cao là fit-content. Các về phía trên 20px với thuộc tính margin-top và overflow: hindden; giúp ẩn đi các thẻ có layout vượt quá phạm vi của Aside.

```
@media only screen and (max-width: 46.25em) {

    .section {
        flex-wrap: wrap;
    }

    .aside, .article [
        width: 100%;
    ]
}
```

Khi trình duyệt có kích thước bé hơn hoặc bằng 46.25em => 740px thì section sẽ thêm thuộc tính flex-wrap: wrap giúp các item sẽ không bị bóp méo mà

tự động xuống hàng khi thẻ cha có kích thước nhỏ hơn tổng kích thước của thẻ con. Aside và Article sẽ được set lại chiều rộng là 100% để vừa với kích thước của thẻ cha.

```
<div class="film-category-container" id="filmCategoryContainer"></div>

.film-category-container {
    width: 97.5%;
    min-height: 300px;
    display: grid;
    grid-template-columns: repeat(var(--amountOfCol), 1fr);
    grid-template-rows: auto;
    margin: 10px auto;
    grid-gap: 15px;
}
```

```
:root {
    --amountOfCol: 4;
    --amountOfCol_category: 6;
    --amountOfCol_user: 3;
}
```

Trong Article có thẻ “film-category-container” là thẻ chính chứa phim trong website. Có chiều rộng là 97.5% so với thẻ cha chứa nó mà chiều cao tối thiểu là 300px. Các item trong thẻ sẽ được dàn layout bởi thuộc tính Display: grid; Số cột sẽ được quyết định bởi biến **amountOfCol** như ở trên là 4 cột với kích thước bằng nhau (1fr). Số hàng sẽ tự động tăng hay giảm theo số item và các item sẽ cách nhau về 4 phía 15px. Margin: 10px auto giúp nó có thể nằm chính giữa theo chiều ngang của thẻ cha và cách phía trên và dưới của nó 1 khoảng 10px.

```
@media only screen and (max-width: 37.5em) {  
    .film-category-container {  
        --amountOfCol: 3;  
    }  
}
```

```
@media only screen and (max-width: 25.875em) {  
    .film-category-container {  
        --amountOfCol: 2;  
    }  
}
```

```
@media only screen and (min-width: 46.25em) and (max-width: 63.9375em) {  
    .film-category-container {  
        --amountOfCol: 3;  
    }  
}
```

Khi màn hình có kích thước chiều rộng từ 63.9375 em (1023px) cho tới 46.25 em (740px) và khi chiều rộng có kích thước từ 37.5 em (600px) thì số cột của “film-category-container” sẽ là 3 cột. Còn khi có kích thước từ 25.875 em (414px) thì số cột là 2.

```
<div class="film-container ${className}">
  <div class="film-view">
    <i class="fas fa-eye"></i> ${film.view}
  </div>
  <div class="film-bookmark" id="film-bookmark">
    <i class=" ${film.bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon"></i>
  </div>
  
  <div class="film-title">
    <p class="main-title">${film.filmName}</p>
    <p class="sub">${film.subTitle}</p>
  </div>
  <div class="mask">
    <i class="far fa-play-circle"></i>
    <p class="info">Mô tả</p>
    <p class="episode">${film.episode}</p>
  </div>
</div>
```

```
.film-container {
  /* display: none; */
  z-index: 1;
  min-width: 33.33%;
  height: 250px;
  position: relative;
  cursor: pointer;
  transition: all 0.2s linear;
}
```



Mỗi bộ phim trong sẽ được thẻ “film-container” bọc lại. Nó sẽ có chiều rộng ít nhất là 33.33%, chiều cao là 250px. Thuộc tính position: relative; giúp chia layout cho các item trong nó. Cursor: pointer; giúp cho con trỏ chuột thành hình bàn tay khi di vào nó. Transition: all 0.2s linear; giúp cho các hiệu ứng chuyển cảnh trong nó chở nên đều và mượt hơn.

```
.film-container:hover .mask {  
    visibility: visible;  
    opacity: 1;  
}
```

Khi hover vào “film-container” thì “mark” (là lớp phủ mờ) sẽ hiện lên. Nó chứa các item để tương tác cũng như thể hiện thông tin của bộ phim cho người dùng thấy như nút play, số tập, mô tả của phim cũng như là phim đã được lưu vào mục ưa thích hay chưa.

```
.mask {  
    width: 100%;  
    height: 100%;  
    background-color: #1a374dd3;  
    position: absolute;  
    top: 0;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    visibility: hidden;  
    opacity: 0;  
    transition: all 0.35s ease;  
    border-radius: 3px;  
}
```

Mark sẽ có chiều dài và chiều rộng vừa với kích thước của thẻ chứa nó. Position: absolute; và top: 0; giùr nó sẽ nằm khớp vs thẻ cha. Visibility: hidden, opacity: 0; giúp ẩn mark đi cho tới khi người dùng hover vào thẻ cha của nó như đã trình bày ở bên trên. Border-radius: 3px; giúp bo tròn góc của nó.

```

<div class="film-top1-container top1view">
    
    <span class="film-top1-name">
        ${film.filmName}
        <button class="watch-btn btn">Xem ngay</button>
    </span>
    <div class="film-view">
        <i class="fas fa-eye"></i> ${film.view}
    </div>
</div>
<div class="film-top-container topview mobile-respon">
    
    <div class="film-top-title">
        <span class="name">${film.filmName}</span>
        <span class="sub"><br>${film.subTitle}</span>
    </span>
        <span class="view">${film.view} lượt xem</span>
    </div>
</div>

```



```

.film-top1-container {
    width: 94%;
    height: 200px;
    margin: 15px auto 0 auto;
    position: relative;
    overflow: hidden;
    cursor: pointer;
}

```

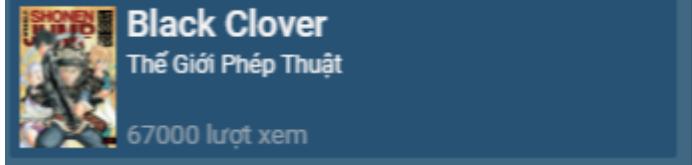
Khi là phim có lượt xem nhiều nhất thì nó sẽ có class "film-top1-container".

Chiều rộng của thẻ là 94% và chiều cao là 200px. Margin: 15px auto 0 auto giúp nó sẽ nằm chính giữa của thẻ cha chứa nó theo chiều ngang và cách những thẻ phía trên nó 15px. Các item trong thẻ sẽ được sắp xếp bằng thuộc tính position: relative;

```
<div class="film-top-container topview">
    
    <div class="film-top-title">
        <span class="name">${film.filmName}</span>
        <span class="sub"><br> ${film.subTitle}</span>
    </span>
    <span class="view">${film.view} lượt xem</span>
    </div>
</div>
```

```
.film-top-container {
    width: 95%;
    height: 80px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    background-color: #275274;
    border-radius: 2px;
    margin: 10px auto;
    cursor: pointer;
    transition: all 0.15s linear;
}
```

```
.film-top-container:hover {
    transform: translate(2px, 0);
}
```



Khi phim là phim từ top 2 lượt xem trở thì sẽ có class là “film-top-container”. Nó sẽ có kích thước bằng với 95% kích thước của thẻ cha và chiều cao là 80px. Các item trong nó sẽ được chia bằng thuộc tính display: flex; Ngoài ra khi hover vào thì nó sẽ dịch chuyển ngang theo chiều trục x 2px nhờ thuộc tính transform: translate(2px, 0);

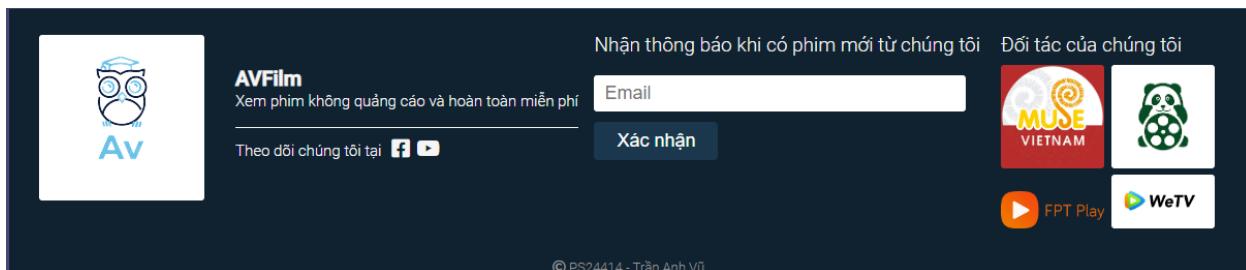
```
<div class="footer">
  <div class="footer-container">

    <div class="footer-leftright">
      <div class="footer-logo">...
      </div>
      <div class="subscribe">
        <p>Nhận thông báo khi có phim mới từ chúng tôi!</p>
        <input type="email" class="input_subscribe" placeholder="Email">
        <button type="button" class="btn_subscribe">Xác nhận</button>
      </div>
      <div class="cooperation">...
      </div>
    </div>

    <span class="copyright">
      <i class="far fa-copyright copyright-icon"></i>
      PS24414 - Trần Anh Vũ
    </span>
  </div>
</div>
```

```
.footer {
  width: 100%;
  height: fit-content;
  min-height: 250px;
  background-color: #102230;
  margin-top: 10px;
  position: relative;
}

.footer-container {
  width: 98%;
  height: 100%;
  margin: 0 auto;
}
```



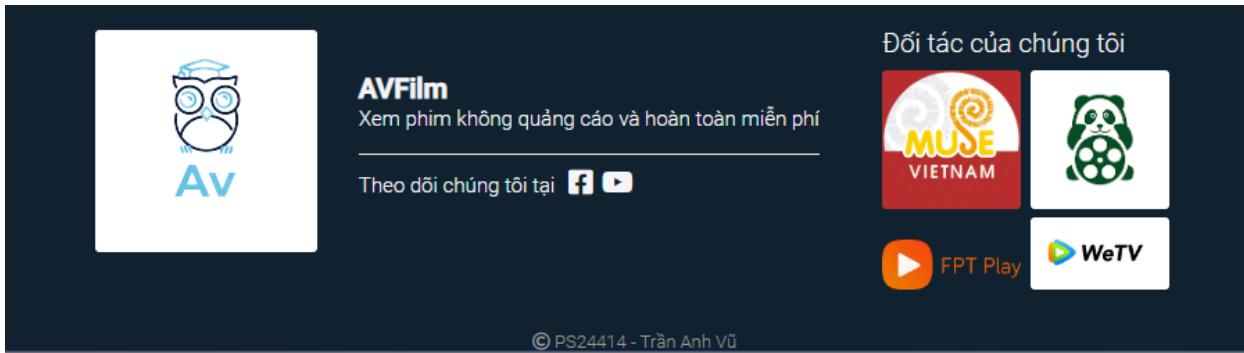
Footer sẽ gồm 3 thành phần chính là phần “footer-logo” (chứa logo website, tiêu đề, slogan, cũng như cách thức liên hệ), “subscribe” (đăng ký việc thông báo khi có tập mới của cũng như thời gian những phim theo dõi).

Thẻ chứa tất cả 3 phần trên là footer-leftright và nó được chứa bọc bởi footer-container”. “footer-container” sẽ có chiều rộng bằng 98% so với thẻ cha và chiều cao là 100% (ở thẻ cha là fit-content, tức là sẽ phụ thuộc vào chiều rộng khi các item khi dàn layout).

```
.footer-leftright {
  width: 100%;
  height: 90%;
  display: flex;
  justify-content: center;
  margin: 0 auto;
```

“footer-leftright” sẽ có chiều rộng là 100% kích thước của thẻ cha chứa nó và chiều rộng là 90px. Các item trong thẻ sẽ được display: flex; căn chỉnh ra giữa với thuộc tính justify-content: center; và thẻ sẽ tự động ở giữa theo chiều ngang so với thẻ cha nhờ thuộc tính margin: 0 auto;

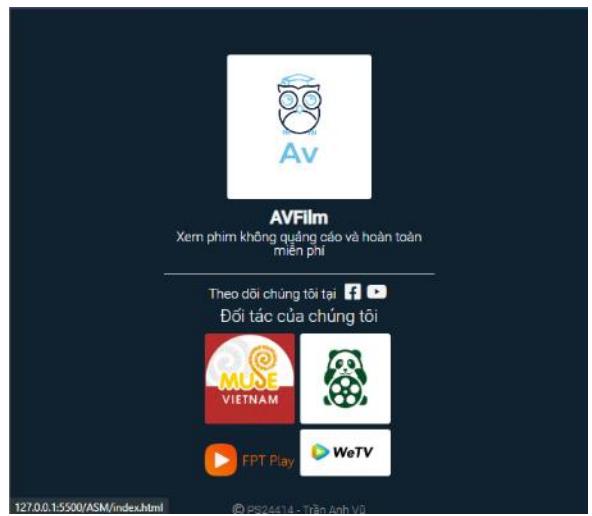
```
@media only screen and (max-width: 64em) {
    .footer-leftright {
        justify-content: space-around;
    }
}
```



Khi màn hình có kích thước từ 64em (1024px) thì thuộc tính justify-content sẽ được đổi thành space-around thay vì center. Và thẻ “subscribe” sẽ được ẩn đi.

```
@media only screen and (max-width: 46.25em) {
    .footer-leftright {
        display: block;
    }
}
```

Khi màn hình có kích thước từ 46.25em (740px) thì “footer-leftright” sẽ có thuộc tính display là block để các item có thể nằm thẳng theo 1 hàng



- Trang danh mục phim:

```
<div class="web-container">
    <!-- Header Mobile -->
    <header class="header-mobile">...
    </header>
    <!-- Header -->
    <header class="header-top">...
    </header>
    <div class="header-center">...
    </div>
    <div class="header-bottom">...
    </div>

    <!-- Section -->
    <section class="section">...
    </section>
    <!-- Footer -->
    <footer class="footer">...
    </footer>
</div>
```

Trang vẫn có cấu trúc như trang chủ với 3 phần chính là header, section và footer. Chỉ có thay đổi lớn nhất là của section

```
<section class="section">
    <div class="article">
        <div class="article-container">
            <div class="film-category-header" id="filmCategoryHeader"></div>
            <div class="film-category-container" id="filmCategoryContainer"></div>
            <!-- <a href="#" class="other-link">
            |   <button class="other-btn btn">Xem thêm</button>
            |</a> -->

            <div class="page">
                <svg enable-background="new 0 0 11 11" viewBox="0 0 11 11" x="0" y="0"
                    class="page-icon icon-left">
                    <g>
                        <path
                            d="m8.5 11c-.1 0-.2 0-.3-.1l-6-5c-.1-.1-.2-.3-.2-.4s.1-.3.2-.4l6-5c .2-.2.5-.1.7.1s.1.5-.1.7l-5.5 4.6 5
                        </path>
                    </g>
                </svg>
                <div class="number-page page-sentence">1</div>
                <div class="number-page">2</div>
                <div class="number-page">3</div>
                <div class="number-page dot">..</div>
                <div class="number-page">12</div>
                <svg enable-background="new 0 0 11 11" viewBox="0 0 11 11" x="0" y="0"
                    class="page-icon icon-right">
                    <g>
                        <path
                            d="m8.5 11c-.1 0-.2 0-.3-.1l-6-5c-.1-.1-.2-.3-.2-.4s.1-.3.2-.4l6-5c .2-.2.5-.1.7.1s.1.5-.1.7l-5.5 4.6 5
                        </path>
                    </g>
                </svg>
            </div>

            <div class="film-category-header">Phim xu hướng</div>
            <div class="film-category-container" id="filmTrendCategory"></div>
            <a href="#" class="other-link">
                <button class="other-btn btn">Xem thêm</button>
            </a>
        </div>
    </div>
```

In 318 Col 1 Spaces:4 UTF-8 CR LF HTML Ⓜ Port

Section sẽ gồm 2 phần là: phim của thể loại mà người dùng đã bấm vào xem và những phim đang là xu hướng.

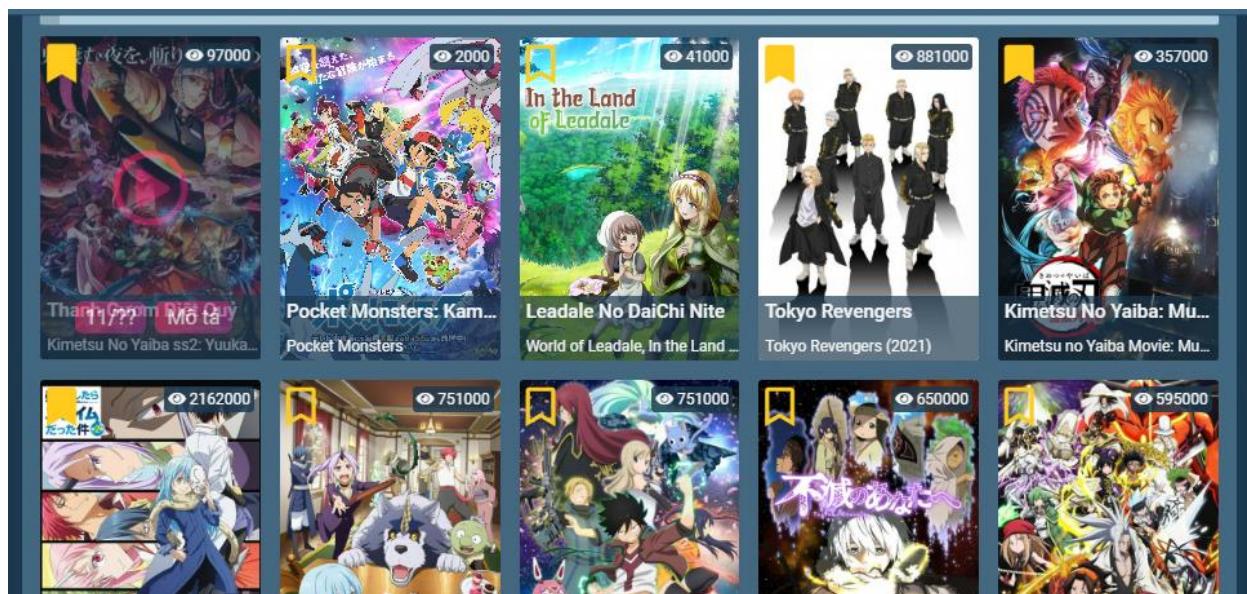
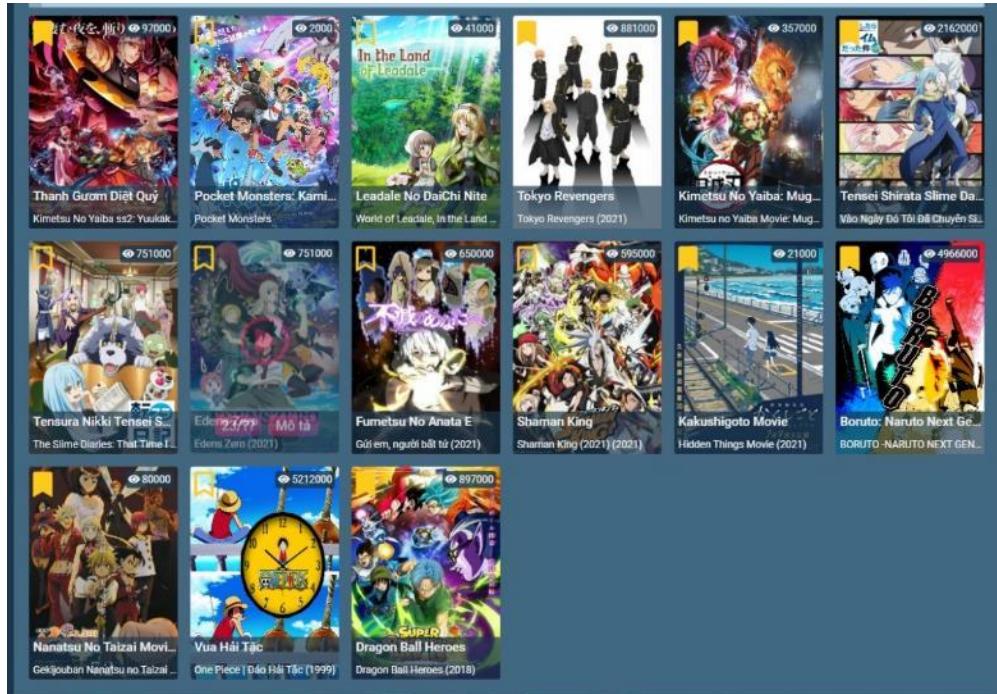
```
<div class="film-category-container" id="filmCategoryContainer"></div>
```

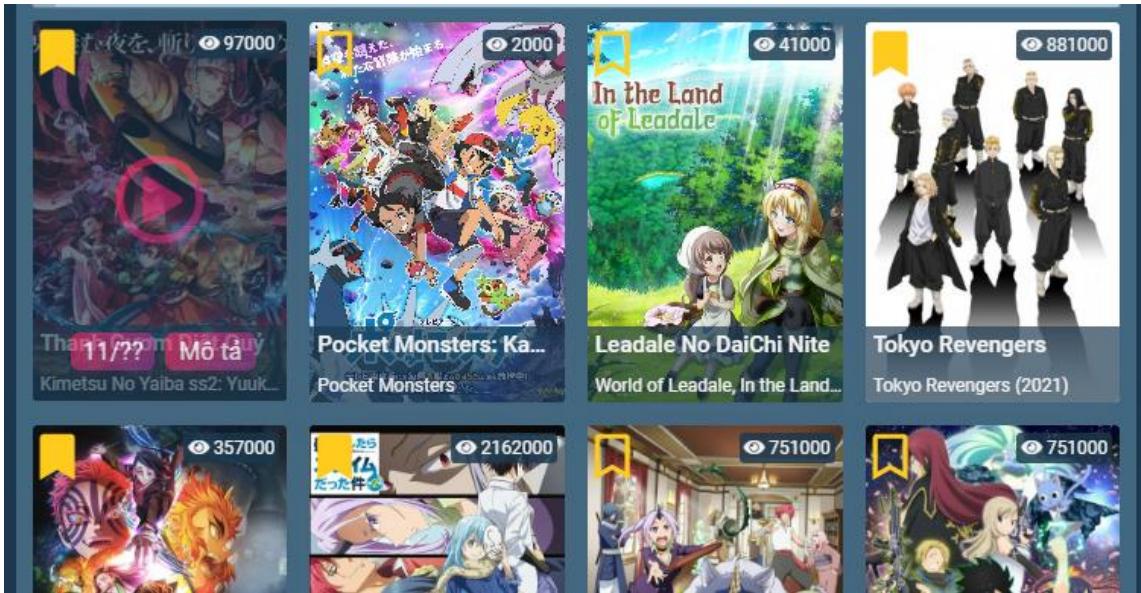
```
.film-category-container {  
    width: 97%;  
    grid-template-columns: repeat(var(--amountOfCol_category), 1fr);  
}
```

Ngoài việc kế thừa các thuộc tính CSS từ trang chủ, “film-category-container” ở trang thể loại sẽ thay đổi chiều rộng bằng 97% so với thẻ cha và số cột của nó sẽ phụ thuộc vào biến **amountOfCol\_category** và kích thước của tất cả các cột là như nhau (1fr).

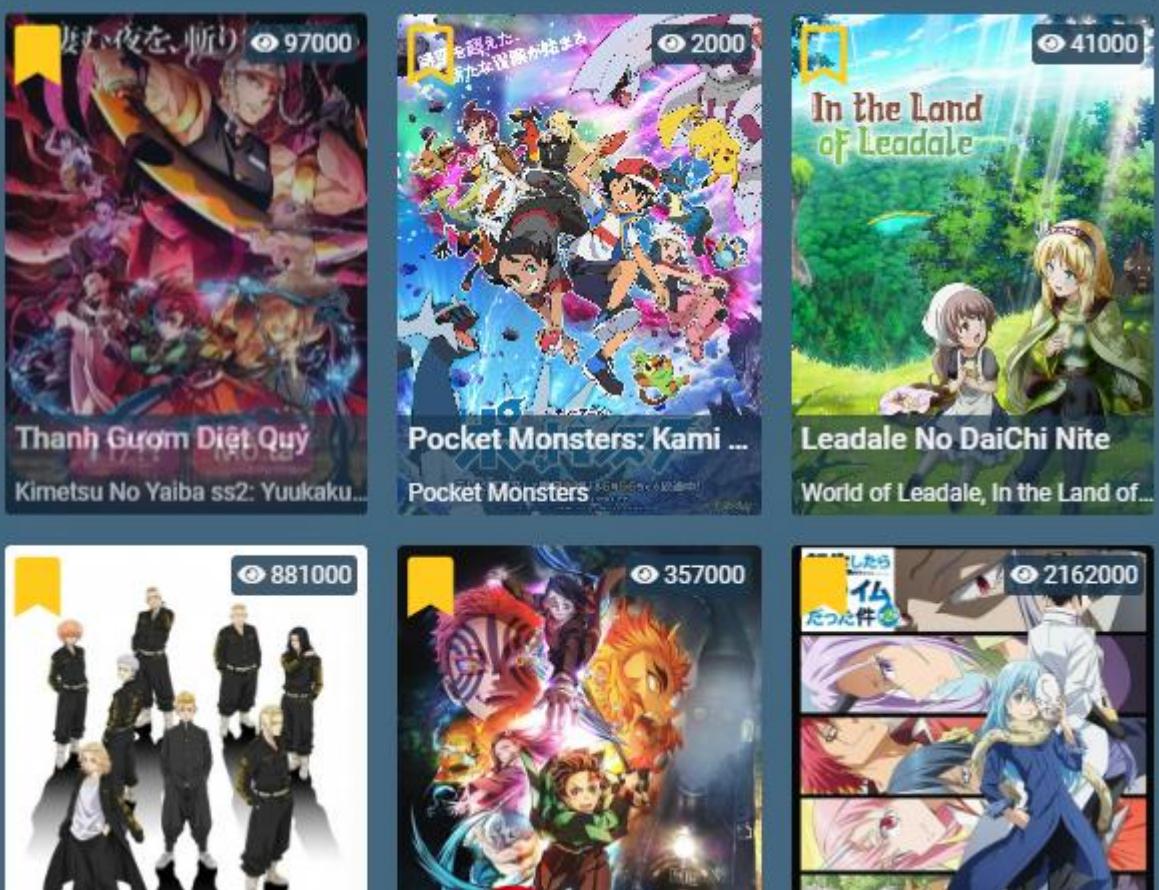
```
@media only screen and (min-width: 51.875em) and (max-width: 62.5em) {  
    .film-category-container {  
        --amountOfCol_category: 5;  
    }  
}  
  
 @media only screen and (min-width: 38.125em) and (max-width: 52em) {  
    .film-category-container {  
        --amountOfCol_category: 4;  
    }  
}  
  
 @media only screen and (min-width: 30em) and (max-width: 38.126em) {  
    .film-category-container {  
        --amountOfCol_category: 3;  
    }  
}  
  
 @media only screen and (max-width: 29.89em) {  
    .film-category-container {  
        --amountOfCol_category: 2;  
    }  
}
```

Khi màn hình có kích thước nằm trong khoảng 51.875em (830px) cho đến 62.5em (1000px) thì số cột của layout sẽ là 5. Khi màn hình có kích thước nằm trong khoảng 38.125em (610px) cho đến 52em (832px) thì số cột của layout sẽ là 4. Khi màn hình có kích thước nằm trong khoảng 30em (480px) cho đến 38.126em (~611px) thì số cột của layout sẽ là 3. Khi màn hình có kích thước bé hơn hoặc bằng 29.89em (478.24px) thì số cột của layout sẽ là 2.





## Shounen



**Shounen**



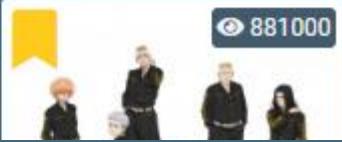
**Thanh Gươm Diệt Quỷ**  
Kimetsu No Yaiba ss2: Yuuka...



**Pocket Monsters: Kami...**  
Pocket Monsters



**In the Land of Leadale**



881000

**Comedy**



Website hiện chưa cập nhật phim của thể loại Comedy. Xin lỗi bạn vì sự bất tiện này.

```

if (container.childElementCount == 0) {
    // set số cột cho "film-category-container" là 1
    container.style.gridTemplateColumns = "1fr";
    // làm trống thẻ phần trang
    pageNumber.innerHTML = '';
    // tạo thẻ img, 1 thẻ p và 1 thẻ div để chứa tất cả dòng thông báo
    const img = document.createElement('img');
    const text = document.createElement('p');
    const containerAlert = document.createElement('div');
    // set class cho thẻ chứa thông báo là "alert-container"
    containerAlert.className = 'alert-container';
    // set chữ hiển thị trong thẻ p
    text.innerText = 'Website hiện chưa cập nhật phim của thể loại ' + category + '. Xin lỗi bạn vì sự bất tiện này.'
    text.classList = 'empty-text';
    // set hình ảnh cho thẻ img
    img.src = 'assets/images/cartoon-owls-sad.png';
    img.classList = 'empty-img';
    // Thêm thẻ p và thẻ img vào trong thẻ "alert-container"
    containerAlert.appendChild(text);
    containerAlert.appendChild(img);
    // thêm "alert-container" vào trong "film-category-container"
    container.appendChild(containerAlert);
}

```

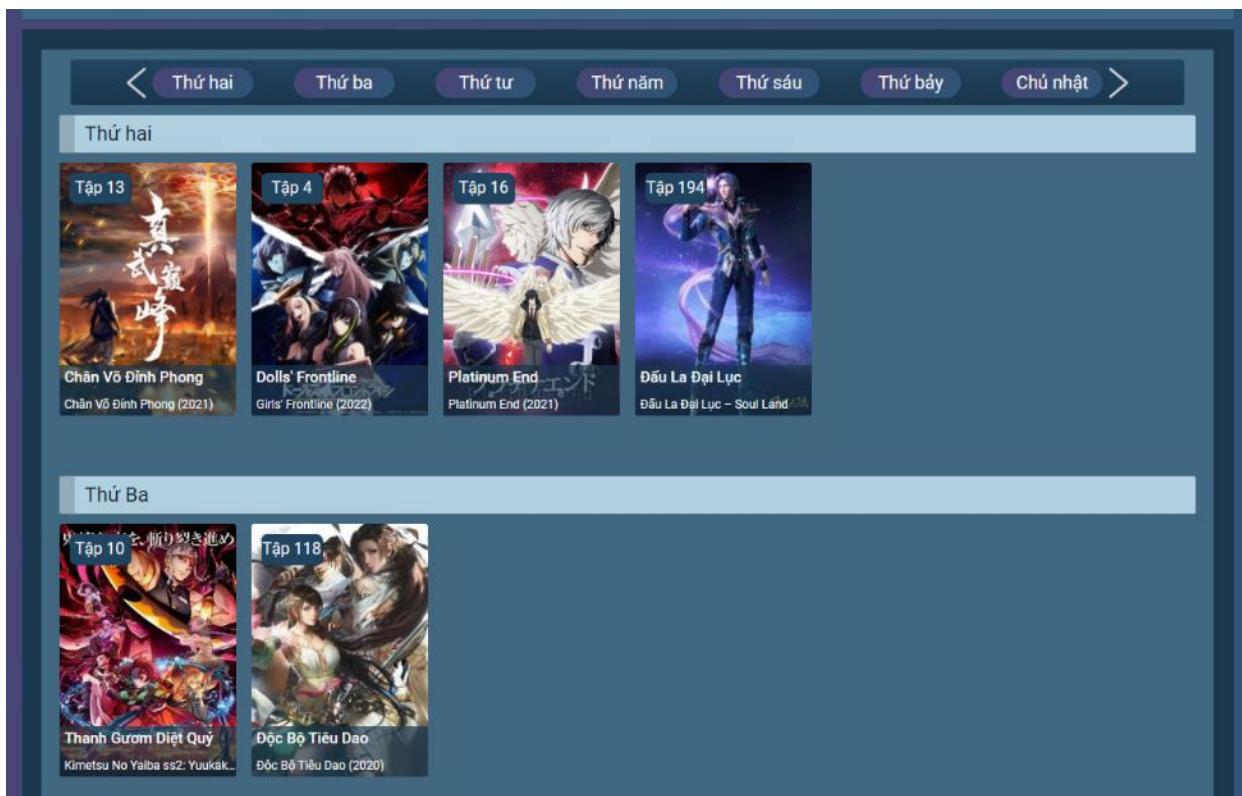
Khi mà thể loại người dùng bấm vào mà website vẫn chưa cập nhật phim thì sẽ hiện thị thông báo rằng phim hiện chưa cập nhật.

```
<div class="film-category-container" id="filmTrendCategory"></div>
```

Phim xu hướng vẫn dữ nguyên layout của “film-category-container” chỉ thay đổi nội dung là các bộ phim bên trong nó.

- **Trang lịch chiếu phim**

Vẫn dữ nguyên phần header và footer từ trang chủ, phần thay đổi chủ yếu là ở section với việc chỉ có article mà không có aside.



Article gồm 2 thành phần chính là showtime (cây hiện thị thời gian những thứ trong tuần có phim ra mắt) và article-container (chứa phim sắp ra mắt theo từng thứ trong tuần)

```
<div class="article">
    <div class="showtime">...
    </div>
    <div class="article-container">...
    </div>
</div>
```

Thứ hai

Thứ ba

Thứ tư

Thứ năm

Thứ sáu

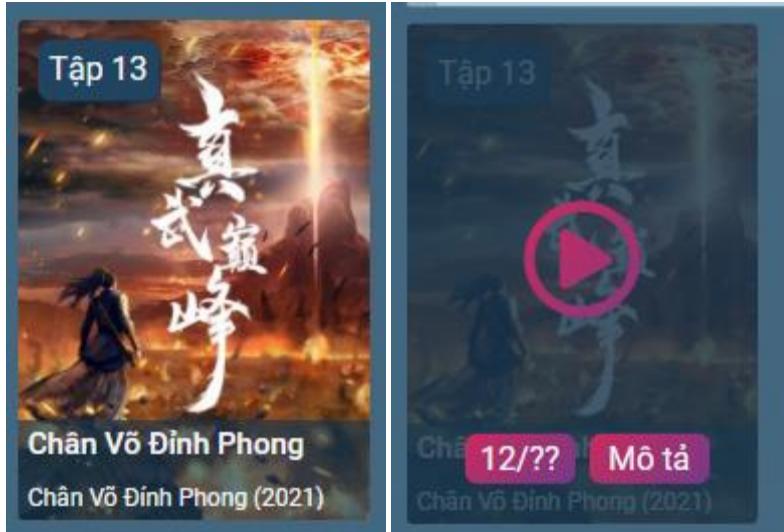
Thứ bảy

Chủ nhật



```
.showtime {  
    position: sticky;  
    top: 109px;  
    display: flex;  
    align-items: center;  
    justify-content: space-evenly;  
    width: 95%;  
    margin: 0 auto;  
    border-radius: 3px;  
    height: 45px;  
    background: linear-gradient(to bottom, #214561, #1A374D);  
    margin-top: 10px;  
    overflow: auto;  
    z-index: 3;  
}
```

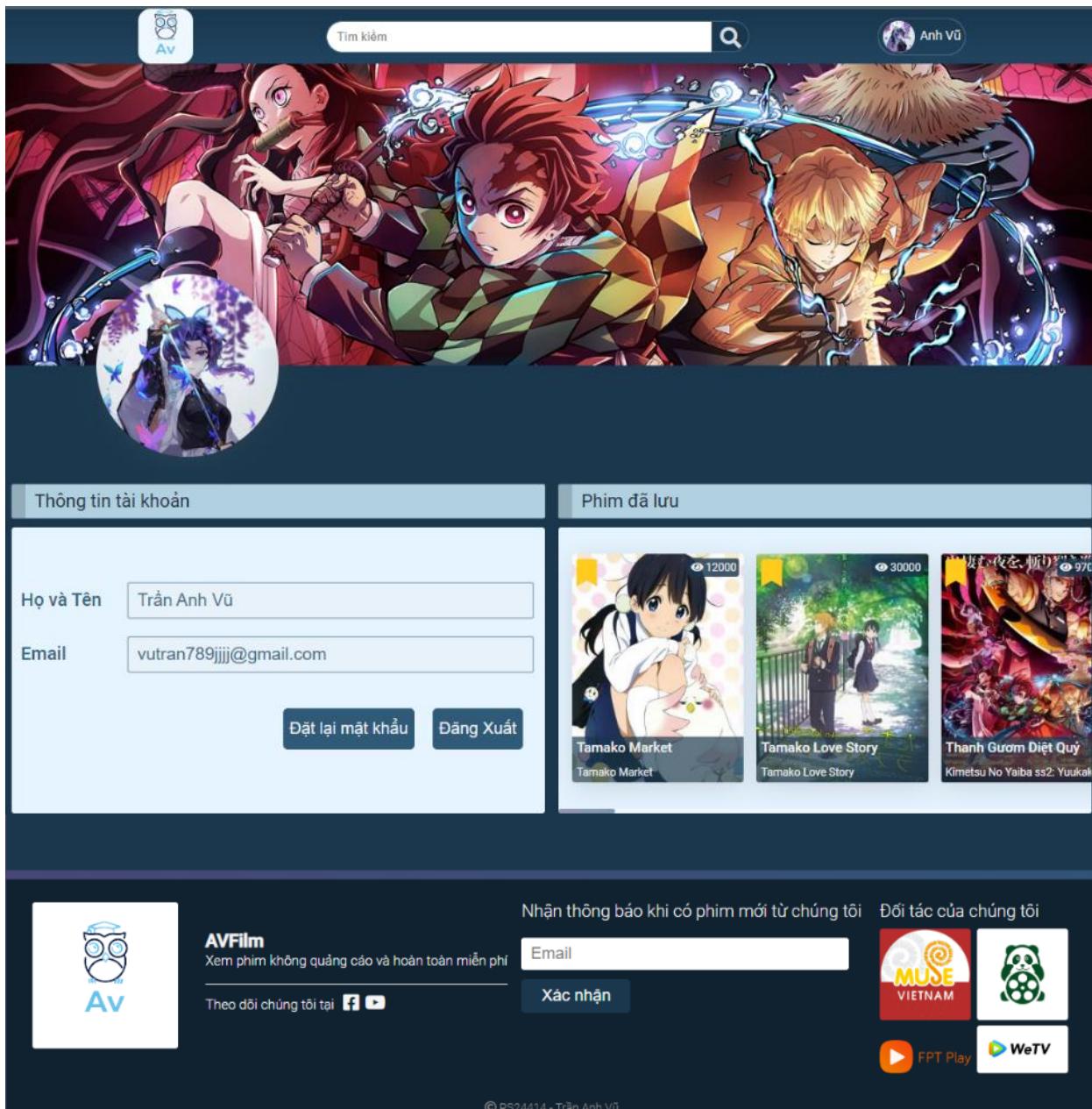
"showtime" sẽ có thuộc tính position: sticky; và top: 109px; giúp nó có thể đứng yên khi nó cách top 109px. Các item trong nó sẽ được thuộc tính display: flex; dàn layout. Các item sẽ cách đều nhau và nằm chính giữa thẻ cha theo chiều dọc. Chiều dài của thẻ là 95% so với thẻ cha và chiều cao là 45px. Background: linear-gradient(to bottom, #214561, #1A374D); giúp thẻ có bacground với màu chuyển. Overflow: auto giúp nó có thể tự sinh ra scroll bar khi các item con vượt quá kích thước thẻ cha.



```
.film-episode {
    box-shadow: 1px 1px 1px #1A374D;
    width: 60px;
    height: 30px;
    color: #f5f5f5;
    line-height: 30px;
    text-align: center;
    position: absolute;
    top: 10px;
    left: 10px;
    border-radius: 6px;
    background-color: #214561;
}
```

Layout của 1 bộ phim sẽ thay đổi. Thay vì có bookmark và lượt xem thì bây giờ chỉ còn thấy số tập sắp ra mắt. Thẻ hiện thị số tập sẽ có class là “film-episode”. Thuộc tính box-shadow giúp độ bóng cho thẻ. Thẻ sẽ có chiều rộng là 60px và chiều cao là 30px với chữ là màu trắng đồng thời căn giữa với text-align: center. Position: absolute giúp điều chỉnh vị trí của nó trong thẻ cha, ở đây là cách phía trên thẻ cha 10px và phía bên trái 10px.

- Trang quản lý tài khoản



Thông tin tài khoản

Họ và Tên: Trần Anh Vũ

Email: vutran789jjjj@gmail.com

Đặt lại mật khẩu      Đăng Xuất

Phim đã lưu

Tamako Market (12000 views)  
Tamako Love Story (30000 views)  
Thanh Gươm Diệt Quỷ (970 views)

AVFilm  
Xem phim không quảng cáo và hoàn toàn miễn phí

Theo dõi chúng tôi tại: [Facebook](#) [YouTube](#)

Nhận thông báo khi có phim mới từ chúng tôi

Email: \_\_\_\_\_

Xác nhận

MUSE VIETNAM

FPT Play

WeTV

© PS24414 - Trần Anh Vũ

Layout vẫn sẽ gồm 3 phần chính là header, section, footer. Trong đó footer dữ nguyên và header-bottom sẽ được thay thế vào đó là avata của user.

```
<div class="mainavt-container">
    
    <i class="fas fa-camera"></i>
</div>
```

```
.mainavt-container {
    width: 200px;
    height: 200px;
    border-radius: 50%;
    position: absolute;
    left: 100px;
    z-index: 2;
    bottom: -100px;
    overflow: hidden;
    box-shadow: rgba(100, 100, 111, 0.2) 0px 7px 29px 0px;
    cursor: pointer;
}
```

"mainavt-container" sẽ có chiều rộng bằng với chiều cao là 200px. Border-radius: 50%; giúp cho thẻ sẽ trở thành hình tròn. Kèm với Overflow: hidden thì những thẻ như img khi có chiều rộng và cao bằng với nó sẽ bị che đi nhưng phần rìa chỉ để lại phần hình tròn. Vị trí của thẻ sử dụng thuộc tính position: absolute để căn.

```
✓ @media only screen and (max-width: 63.9375em) {
    ✓ .mainavt-container {
        display: none;
    }
}
```

Khi màn hình có kích thước từ dưới hoặc bằng 63.9375em (1023px) thì thẻ sẽ ẩn đi với thuộc tính display: none;

```

<div class="section">
    <div class="section-userpage">
        <div class="info-container">
            <div class="film-category-header">Thông tin tài khoản</div>
            <div class="info-user">
                <span class="fullname">Họ và Tên</span>
                <input type="text" id="fullname" disabled>
                <br>
                <span class="email">Email</span>
                <input type="text" id="email" disabled>
                <div class="btn-group">
                    <button id="btn_resetPass">Đặt lại mật khẩu</button>
                    <button id="btnLogTowWayInfo">Đăng xuất</button>
                </div>
            </div>
        </div>
        <div class="bookmark-container">
            <div class="film-category-header">Phim đã lưu</div>
            <div class="bookmark" id="filmBookmarkContainer">
                <i class="previous fas fa-angle-left" id="btnPrevious"></i>
                <i class="next fas fa-angle-right" id="btnNext"></i>
            </div>
        </div>
    </div>
</div>

```

Thông tin tài khoản

Họ và Tên	Trần Anh Vũ
Email	vutran789jjjj@gmail.com

[Đặt lại mật khẩu](#)

[Đăng Xuất](#)

Phim đã lưu



Tamako Market  
12000



Tamako Love Story  
30000



Thanh Gươm Diệt Quý  
970

```
.section-userpage {
    width: 100%;
    height: fit-content;
    background-color: #1A374D;
    display: flex;
    justify-content: space-evenly;
    padding-bottom: 20px;
}
```

```
.info-container, .bookmark-container {
    width: 50%;
    height: 100%;
    padding-top: 120px;
}
```

Thẻ “section-userpage” là thẻ chứa 2 thành phần chính là thẻ “info-container” và “bookmark-container”. Nó có chiều rộng là 100% so với thẻ cha của nó và chiều cao là tùy thuộc vào item trong nó. Các item sẽ được sắp xếp bởi thuộc tính display: flex; và căn giữa với justify-content: space-evenly;

“info-container” chứa thông tin cơ bản của người dùng như Họ và Tên, Email và 2 nút: đăng xuất(khi đã đăng nhập và đổi thành đăng nhập khi chưa đăng nhập). Thẻ “bookmark-containier” chứa những bộ phim mà người dùng đã đánh dấu. Cả hai đều có chiều rộng bằng 50% thẻ cha của nó và chiều cao là 100% chiều cao của thẻ cha. Padding-top: 120px giúp nó đệm thêm 1 lớp 120px phía trên.

```
<div class="bookmark" id="filmBookmarkContainer">
    <i class="previous fas fa-angle-left" id="btnPrevious"></i>
    <i class="next fas fa-angle-right" id="btnNext"></i>
</div>
```

```
.bookmark {
    width: 97.5%;
    height: 313px;
    margin: 0 auto;
    border-radius: 3px;
    background-color: #E5F4FF;
    padding: 24px 10px;
    overflow-x: scroll;
    white-space: nowrap;
    scroll-behavior: smooth;
}

.bookmark .film-container {
    margin: 5px;
    width: 180px;
    display: inline-block;
}
```

“bookmark” là thẻ chính chứa các phim đã lưu. Nó chiều rộng là 97.5% so với thẻ cha chứa nó. Thuộc tính overflow-x: scroll; giúp nó chỉ hiện thị thanh scroll bar ngang. Scroll-behavior: smooth; giúp cho lúc cuộn scroll bằng code javascripts chở nên mượt hơn

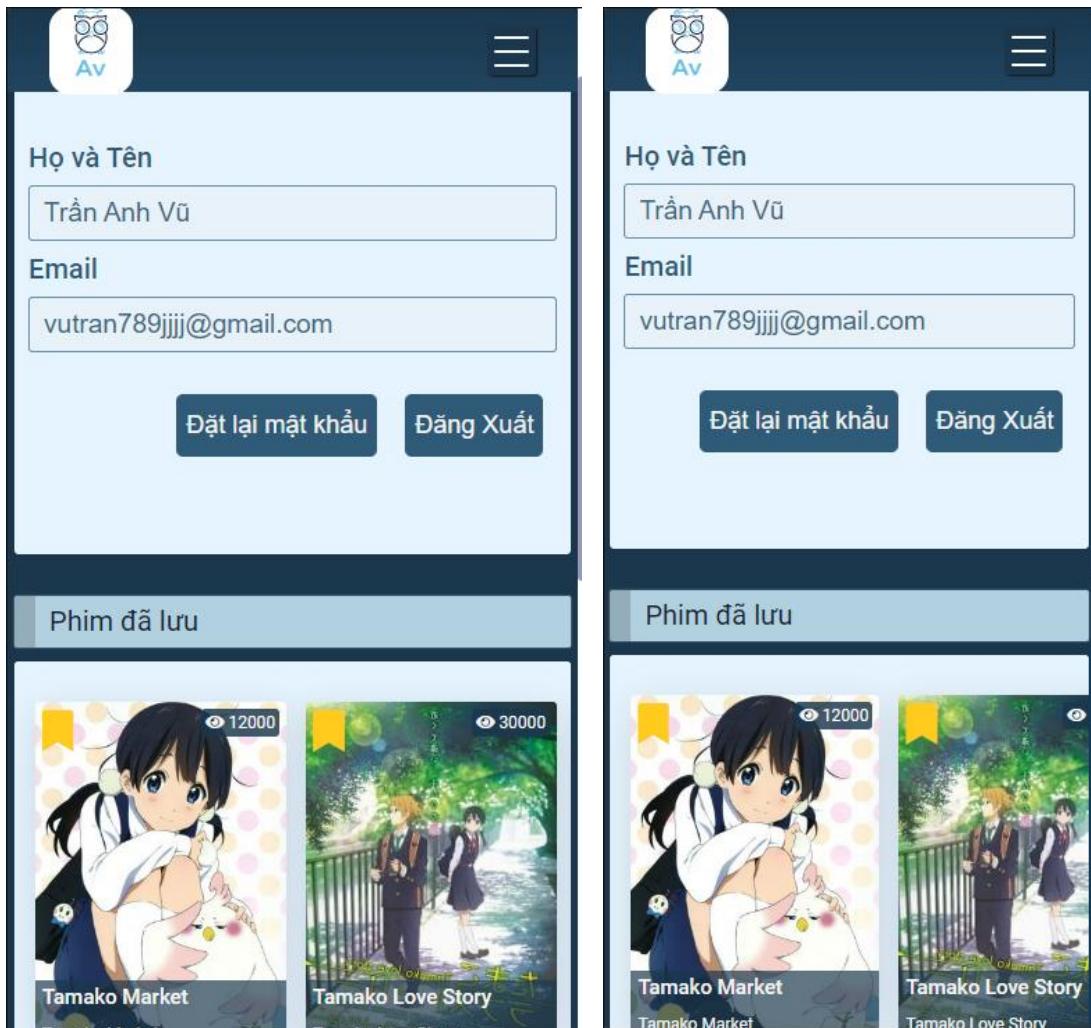
“film-container” cũng được tùy biến với thuộc tính display: inline-block để biến nó thành 1 thẻ nội tuyến. Và chiều rộng của nó là 180px đồng thời các đều xung quanh 1 khoảng là 5px.

```

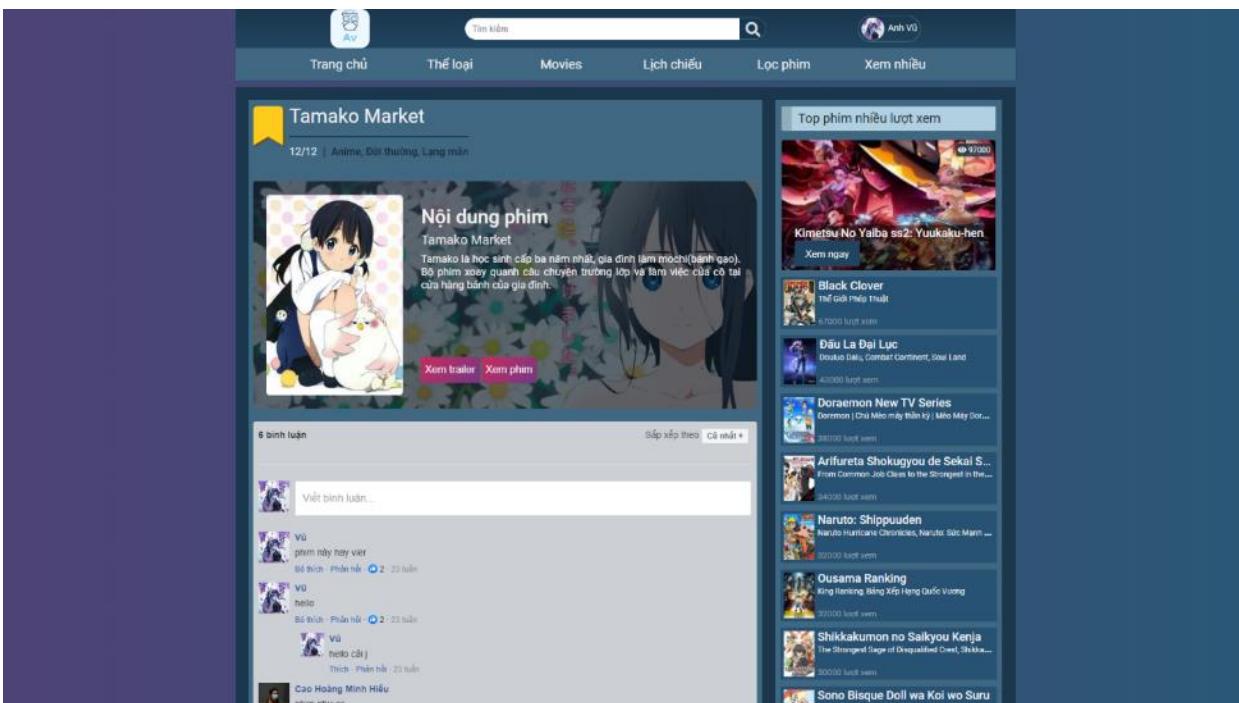
2
3 @media only screen and (max-width: 25.875em) {
4
5     /* for user page */
6     .section-userpage {
7         flex-direction: column;
8     }
9
10    .bookmark-container, .info-container {
11        width: 100%;
12    }

```

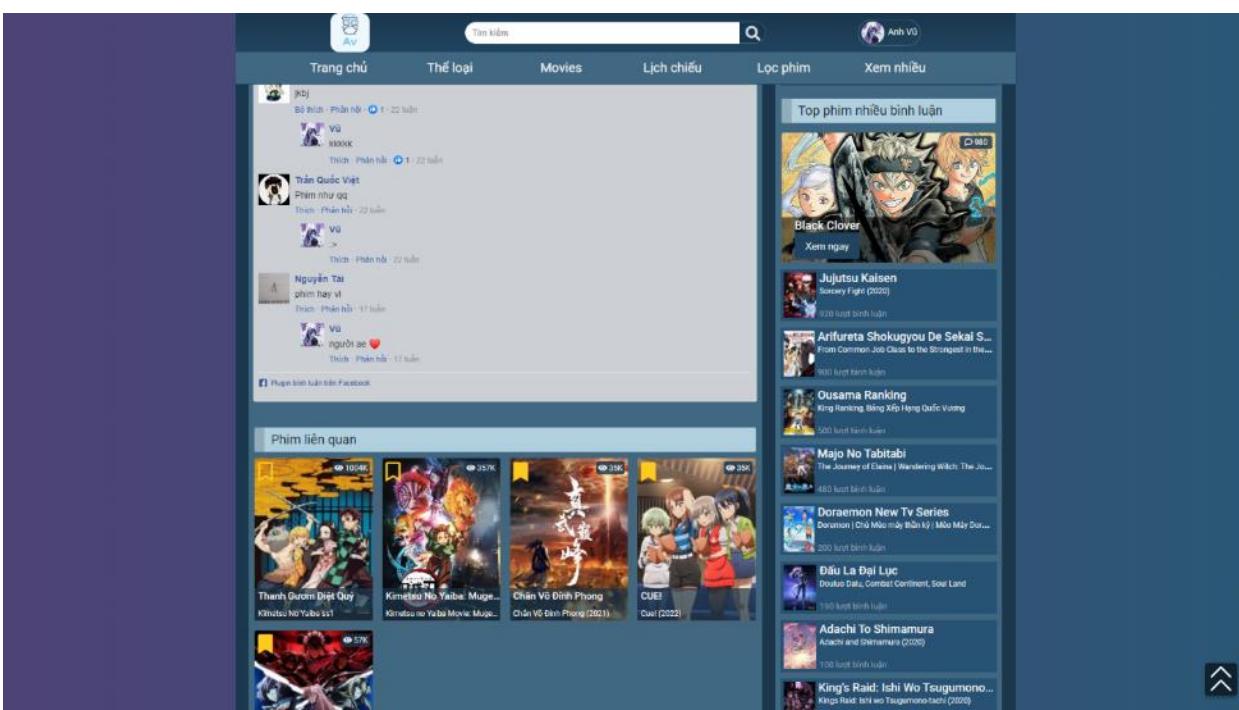
Khi màn hình có kích thước từ 25.875em (414px) thì “section-userpage” sẽ đổi chiều xếp các item theo chiều cột (dọc). đồng thời kích thước của “bookmark-container” và “info-container” sẽ có chiều rộng là 100% kích thước thẻ cha.



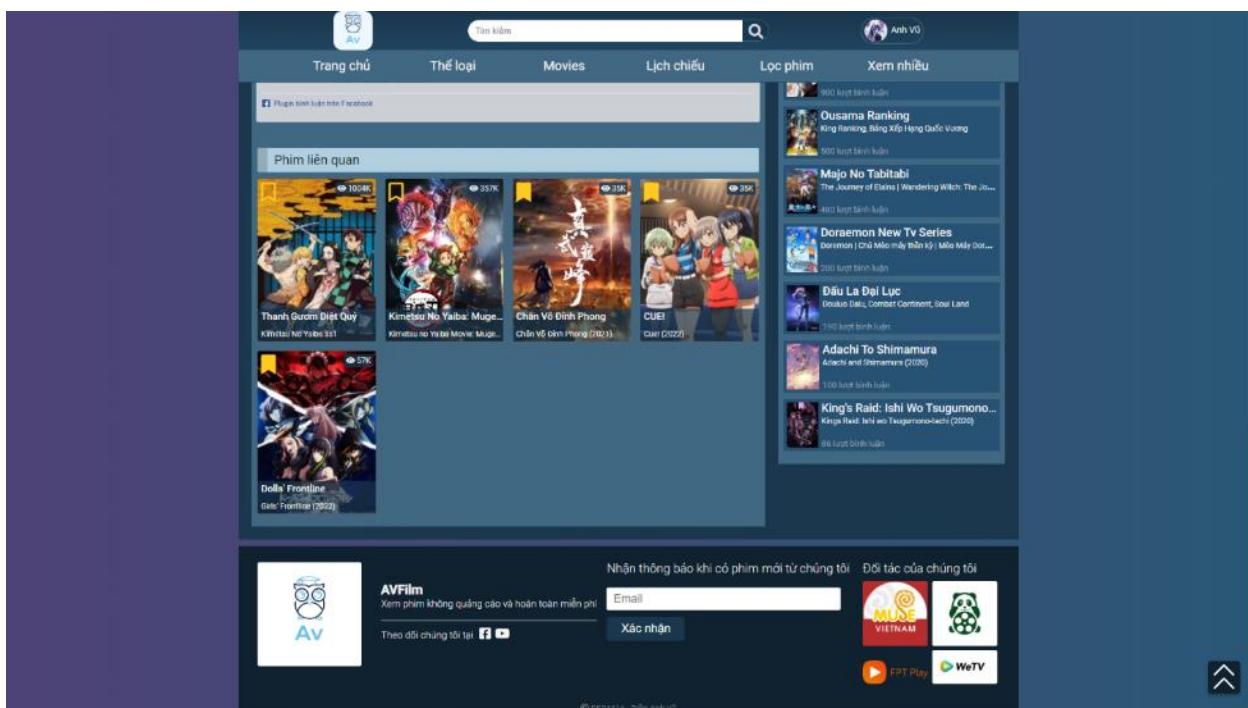
- Trang chi tiết phim:



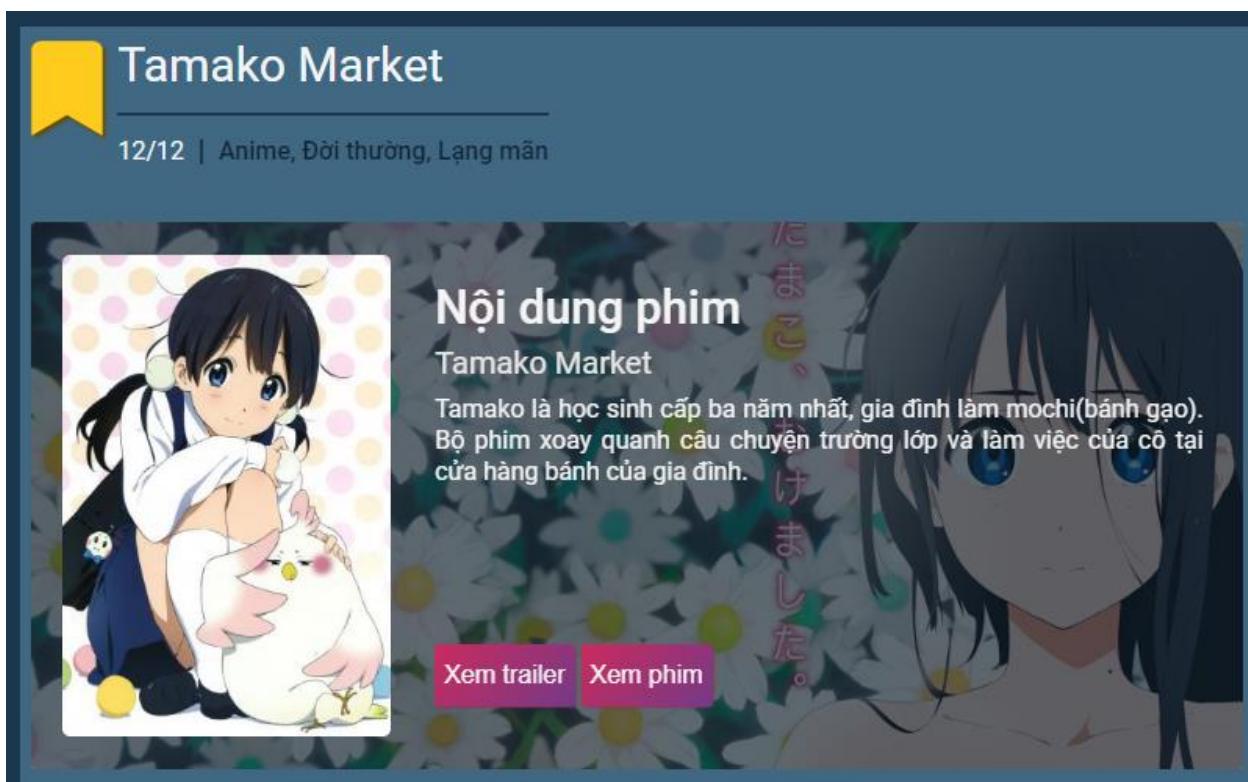
The screenshot shows a movie detail page for 'Tamako Market'. At the top, there's a banner with two anime girls. Below it, the movie title 'Tamako Market' is displayed with a small thumbnail. A summary text describes Tamako as a 12-year-old girl who loves her family and works part-time at a bakery. There are buttons for 'Xem trailer' and 'Xem phim'. On the left, a sidebar shows user reviews and a comment input field. On the right, a sidebar lists 'Top phim nhiều lượt xem' (most viewed movies) with thumbnails and titles like 'Kimetsu No Yaiba ss2: Yuukaku-hen', 'Black Clover', 'Doraemon New TV Series', etc.

The screenshot shows a movie detail page for 'Black Clover'. The main content area displays the movie's cover art, title, and a summary. Below this, there's a section for 'Phim liên quan' (related movies) showing thumbnails for 'Thanh Gươm Diệt Quỷ', 'Kimetsu No Yaiba: Mugen Train', 'Chân Võ Định Phong', and 'CUE!'. On the right, a sidebar lists 'Top phim nhiều bình luận' (most commented movies) with thumbnails and titles like 'Black Clover', 'Jujutsu Kaisen', 'Arifureta Shokugyou De Sekai S...', etc.



Về mặt giao diện thì Header sẽ không còn banner để có thêm không gian cho phần phim chi tiết. Cấu trúc của header cũng không khác so với trang chủ. Thay đổi lớn nhất là về phần Nội dung Container.



```

<div class="film-detail">
    <div class="film-title-pr">
        <div class="film-detail-bookmark">
            <i class="${bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon-detail"></i>
        </div>
        <div class="title-pr">
            <div class="name-film-pr">
                ${filmName}
            </div>
            <div class="film-category__epsoide">
                <p class="film-epsoide-pr">${episode}</p>
                <div class="film-category-pr">${type}</div>
            </div>
        </div>
    </div>
    <div class="film-img-pr" style="background-image: url('${backgroundImg}');">
        <div class="thumbnail">
            
        </div>
        <div class="dec">
            <p class="title-dec">Nội dung phim</p>
            <a href="">
                <p class="name-film-dec">${filmName}</p>
            </a>
            <p class="article-dec">${desc}</p>
            <button class="trailer"><a target="_blank" rel="noopener noreferrer" href="${urlVideo}">Xem trailer</a></button>
            <button class="choice-epsoide" id="watchFilm">Xem phim</button>
        </div>
    </div>
    <div class="comment-field">
        <div class="fb-comments" data-href="https://travux.github.io/AVFilm-demo/film-detail.html"
            data-width="100%" data-numposts="10">
        </div>
    </div>
</div>

```

```

.film-detail {
    width: 98%;
    height: max-content;
    background-color: #406882;
    padding-top: 10px;
    color: #f5f5f5;
    margin: 0 auto;
}

```

Thẻ chứa chi tiết phim sẽ có chiều rộng là 98% so với thẻ cha của nó và chiều cao sẽ phụ thuộc vào còn mà nó chứa.

```

.film-title-pr {
    display: flex;
    width: 100%;
    height: 90px;
}

.film-detail-bookmark {
    font-size: 60px;
    color: #FFCC1D;
    cursor: pointer;
    padding-right: 10px;
    filter: drop-shadow(2px 2px 1px #634d03);
}

.bookmark-icon-detail {}

.title-pr {
    font-size: 30px;
}

.name-film-pr {
    width: 100%;
    border-bottom: 2px solid #1a374d;
    padding-bottom: 15px;
}

.film-category__epsoide {
    display: flex;
    font-size: 16px;
    padding-top: 15px;
}

.film-epsoide-pr {
    padding-right: 10px;
    border-right: 2px solid #1a374d;
}

```

```

.film-category-pr {
    padding-left: 10px;
    transition: all 0.18s linear;
    color: #0f1e2b;
    font-weight: 500;
    filter: brightness(1.2);
    cursor: pointer;
}

.film-category-pr a:hover {
    filter: brightness(1.2);
}

.film-category-pr>a {
    transition: all 0.18s linear;
    color: #0f1e2b;
    font-weight: 500;
}

.film-img-pr {
    margin-top: 25px;
    border-radius: 3px;
    background-repeat: no-repeat;
    background-size: cover;
    background-position: 50% 50%;
    min-height: 350px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    position: relative;
    border-radius: 3px;
}

```

Đây là những thuộc tính giúp thẻ con để có thể tạo ra được layout trên hình.  
Trong đó có nhưng thuộc tính mới như `filter: drop-shadow(2px 2px 1px #634d03);`  
Để có thể tạo ra bóng cho biểu tượng đánh dấu.  
Và `filter: brightness(1.2);` Để có thể tăng độ  
Dòng thẻ loại phim khi hover vào.



Nút xem phim sẽ có tác dụng đưa người dùng tới trang xem phim và nút xem trailer  
giúp người dùng có thể xem được trailer của bộ phim hiện tại.

6 bình luận

Sắp xếp theo Cũ nhất

 **Vũ**  
phim này hay vler  
Bỏ thích · Phản hồi · 2 · 23 tuần

 **Vũ**  
hello  
Bỏ thích · Phản hồi · 2 · 23 tuần

 **Vũ**  
hello cái j  
Thích · Phản hồi · 23 tuần

 **Cao Hoàng Minh Hiếu**  
phim như cc  
Thích · Phản hồi · 22 tuần

 **Vũ**  
>> thua anh luôn  
Thích · Phản hồi · 1 · 22 tuần

 **Poppy Hoàng Bảo**  
jkbj  
Bỏ thích · Phản hồi · 1 · 22 tuần

 **Vũ**  
kkkk  
Thích · Phản hồi · 1 · 22 tuần

 **Trần Quốc Việt**  
Phim như qq  
Thích · Phản hồi · 22 tuần

```
<div class="comment-field">
    <div class="fb-comments" data-href="https://travux.github.io/AVFilm-demo/film-detail.html"
        | data-width="100%" data-numposts="10">
    </div>
</div>
```

```

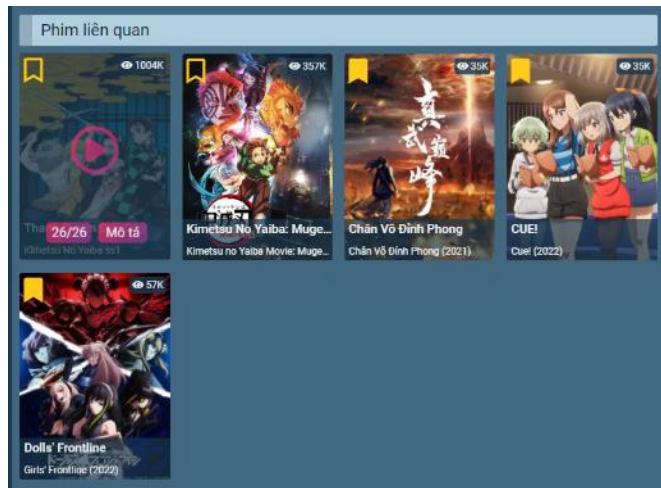
.comment-field {
    background-color: #cccd0d5;
    margin-top: 20px;
    min-height: 80px;
    height: max-content;
    padding-bottom: 10px;
    width: 100%;
    border-radius: 3px;
}

.comment-header {
    width: 98%;
    margin: 0 auto;
    height: 70px;
    display: flex;
    justify-content: space-between;
    padding: 10px 10px;
    color: #1a374d;
    font-weight: 500;
    border-bottom: 2px solid white;
}

.comment {
    height: fit-content;
    font-size: 18px;
    border-bottom: 2px solid currentColor;
}

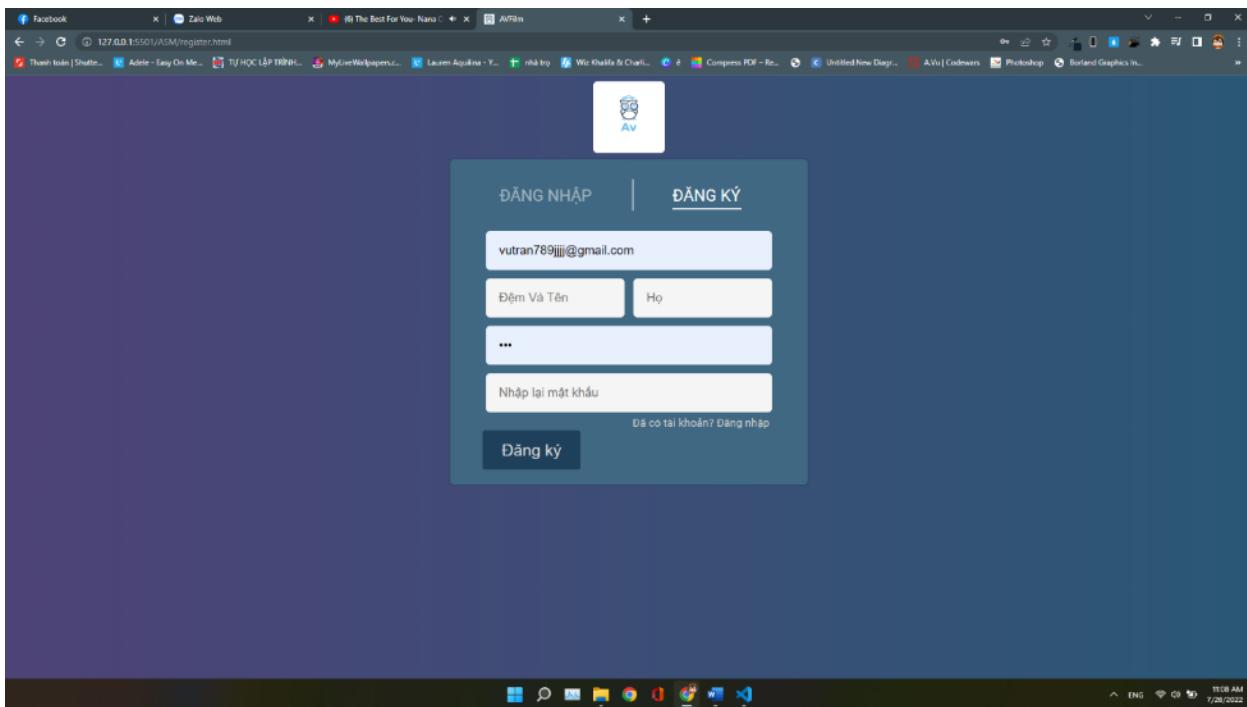
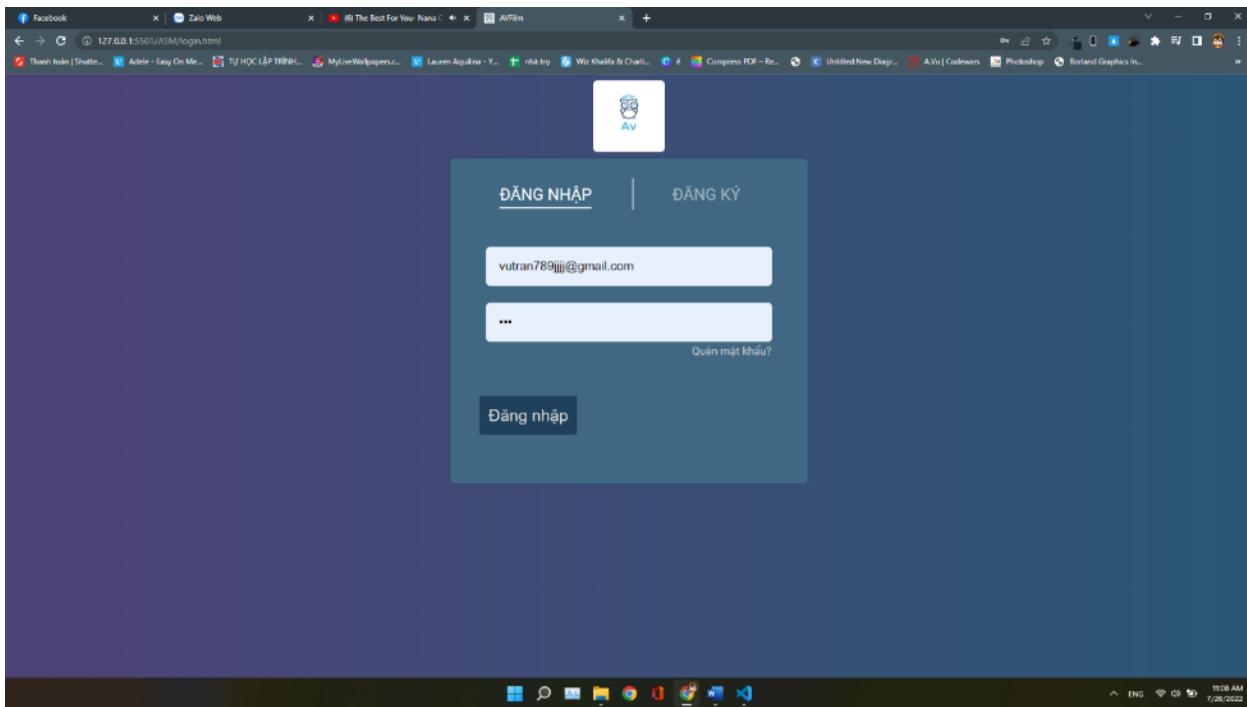
```

Phần comment của phim được sử dụng “**Comment Graph API**” của Facebook để dựng lên dao diện . Thẻ chứa nó sẽ là “comment-field” có kích thước chiều ngang là 100% so với thẻ cha chứa nó và chiều cao sẽ phụ thuộc vào những thẻ con chứa trong nó.



Phần phim liên quan sẽ hiện thị những phim cùng thể loại và layout sẽ tương tự như layout phần “main” của trang chủ.

- Trang đăng ký đăng nhập



```
<div class="web-container">
    <a href="index.html"></a>
    <div class="form login-res">
        <form action="" method="post" enctype="multipart/form-data" class="login-form">
            <div class="form-title">
                <a href="login.html"><span class="login">Đăng Nhập</span></a>
                <a href="register.html"><span class="register">Đăng ký</span></a>
            </div>
            <div class="input">
                <input id="email" type="text" class="input-bar" placeholder="Email" required>
                <input id="password" type="password" class="input-bar pass" placeholder="Mật khẩu" required>
            </div>
            <span class="forgot-password">Quên mật khẩu?</span>
            <button id="btnLogin" type="button" class="login-btn">Đăng nhập</button>
        </form>
        <div class="warning">Điền đầy đủ Email và Password</div>
        <!-- <div class="warning active">Đăng nhập thành công!. Sẽ chuyển tới trang chính trong 4s</div> -->
    </div>
</div>
```

Trang đăng ký/ đăng nhập sẽ có layout khác hoàn toàn so với Trang chủ, Trang chi tiết phim cũng như các trang còn lại.

```
<div class="web-container">
    <a href="index.html"></a>
    <div class="form login-res">
        <form action="" method="post" enctype="multipart/form-data" class="login-form">
            <div class="form-title">
                <a href="login.html"><span class="login">Đăng Nhập</span></a>
                <a href="register.html"><span class="register">Đăng ký</span></a>
            </div>
            <div class="input">
                <input id="email" type="text" class="input-bar" placeholder="Email" required>
                <input id="password" type="password" class="input-bar pass" placeholder="Mật khẩu" required>
            </div>
            <span class="forgot-password">Quên mật khẩu?</span>
            <button id="btnLogin" type="button" class="login-btn">Đăng nhập</button>
        </form>
        <div class="warning">Điền đầy đủ Email và Password</div>
        <!-- <div class="warning active">Đăng nhập thành công!. Sẽ chuyển tới trang chính trong 4s</div> -->
    </div>
</div>
```

Trang đăng nhập sẽ có 2 thẻ input giúp người dùng có thể nhập email và mật khẩu để có thể đăng nhập vào website.

```
.form {
    width: 550px;
    height: 500px;
    background-color: #406882;
    margin: 0 auto;
    border-radius: 5px;
    box-shadow: 0px 0px 2px #1A374D;
}
```

```
.web-container {
    width: min(100%, 1200px);
    margin: 0 auto;
}
```

Form Login sẽ được chứa bởi thẻ có class “form” có chiều rộng là 550px chiều cao là 500px và margin: 0 auto; để có thể căn giữa nó so với thẻ cha là “web-container”.

```
.logo {
    margin: 10px auto 10px;
    display: block;
    width: 110px;
    border-radius: 6px;
}

.login-form {
    width: 100%;
    height: 100%;
    position: relative;
}

.form-title {
    display: flex;
    height: 50px;
    width: 90%;
    position: absolute;
    top: 30px;
    left: 23px;
    justify-content: space-around;
}

.form-title a {
    width: 142px;
    text-align: center;
    display: block;
    text-decoration: none;
}

.form-title span {
    cursor: pointer;
    line-height: 50px;
    width: 150px;
    height: 100%;
    text-align: center;
    font-size: 25px;
    font-weight: 400;
    text-transform: uppercase;
}

.login {
    color: white;
    /* position: absolute; */
    left: 56px;
}

.login:hover {
    color: #rgb(216, 214, 214);
}

.login::after {
    display: block;
    content: "";
    height: 2px;
    width: 141px;
    position: absolute;
    background-color: #f5f5f5;
    bottom: 3px;
    left: 53px;
}

.form-title::before {
    display: block;
    content: "";
    height: 49px;
    width: 2px;
    position: absolute;
    background-color: #f5f5f5;
    left: 50%;
    transform: translateX(calc(-50% + 10px));
}

.register {
    color: white;
    opacity: 0.5;
    /* position: absolute; */
    right: 70px;
}
```

Trên đây là những thuộc tính CSS giúp cho trang Login có layout như trên website. Có những thuộc tính mới như transform: translateX(calc(-50% + 10px)); Giúp cho đường ngăn cách giữa 2 tiêu đề “Đăng ký”, “Đăng nhập”.

### Code JavaScript của trang

```
import { pathnameLoginPage } from "./constants.js";
// Nút login trong form
const btnLogin = document.querySelector("#btnLogin");
// Thẻ thông báo trong form
const warning = document.querySelector(".warning");
var indexUser;
// biến giúp ghi nhận trạng thái đăng nhập của user
var loginSuccess = false;
// Lấy list user được lưu trữ trong LocalStorage
var listUser = localStorage.getItem("listUser");
if (listUser == null) {
  localStorage.setItem("listUser", JSON.stringify([]));
  listUser = localStorage.getItem("listUser");
}
// Thẻ input nhập email trong form
var email = document.getElementById("email");
// Thẻ input nhập password trong form
var password = document.getElementById("password");
// Bắt sự kiện cho nút login. Nếu khi bấm vào nút bấm mà người dùng vẫn chưa nhập thông tin đầy đủ cũng như nhập thông tin
// tài khoản mật khẩu sai thì sẽ không được chấp nhận và đưa ra thông báo "Email hoặc password chưa đúng"
// Nếu người dùng nhập đúng tài khoản và mật khẩu thì chuyển tới trang chủ sau 4s và đưa ra thông báo "Đăng nhập thành công!. Sẽ chuyển tới trang chính trong 4s"
btnLogin.addEventListener("click", () => [
  if (email.value.length > 0 && password.value.length > 0) {
    console.log(JSON.parse(listUser));
    let tempEmail = email.value;
    let tempPassword = password.value;
    JSON.parse(listUser).map((user, index) => {
      console.log(user.email, user.password);
      if (user.email == tempEmail && user.password == tempPassword) {
        console.log('đúng');
        loginSuccess = true;
        localStorage.setItem('indexCurrentUser', index);
        warning.innerText = 'Đăng nhập thành công!. Sẽ chuyển tới trang chính trong 4s';
        console.log(warning.innerText);
        warning.style.backgroundColor = '#A0D995';
        warning.style.borderColor = "#6CC4A1";
        warning.classList = "warning active";
        setTimeout(() => {
          location.pathname = pathnameLoginPage;
          return;
        }, 4000);
      }
    })
  }
])
```

```
<div class="web-container">
    <a href="index.html"></a>
    <form action="" class="form register-res">
        <div class="form-title">
            <a href="login.html"><span class="login">Đăng Nhập</span></a>
            <a href="register.html"><span class="register">Đăng ký</span></a>
        </div>
        <div class="gr-input">
            <input id="email" type="email" class="input-email" placeholder="Email" required>
            <div class="name">
                <input id="firstName" type="text" class="input-name" placeholder="Đệm và Tên" required>
                <input id="lastName" type="text" class="input-name" placeholder="Họ" required>
            </div>
            <input id="password" type="password" class="input-pass" placeholder="Mật khẩu" required>
            <input id="rePassword" type="password" class="input-pass" placeholder="Nhập lại mật khẩu" required>
        </div>
        <span class="have-account">Đã có tài khoản? <a href="login.html">Đăng nhập</a></span>
        <button id="btnRegister" type="button" class="register-btn">Đăng ký</button>
    </form>
    <div class="warning">Password đã nhập không trùng với bên trên</div>
</div>
```

Trang đăng ký sẽ có 5 thẻ input. Một thẻ input để có thẻ nhập email, hai thẻ để nhập thông tin cá nhân (họ và tên) và hai thẻ để có thể nhập mật khẩu cũng như nhập lại mật khẩu.

```
.form {
    width: 550px;
    height: 500px;
    background-color: #406882;
    margin: 0 auto;
    border-radius: 5px;
    box-shadow: 0px 0px 2px #1A374D;
    position: relative;
}
```

Form đăng ký cũng có kích thước tưởng tự như trang đăng nhập. Với chiều rộng là 550px, chiều cao là 500px. Nhưng có thêm 1 thuộc tính mới là position: relative để giúp dàn layout cho những thẻ con trong nó.

```

.login:hover {
    opacity: 1;
}

.register {
    color: white;
    /* position: absolute; */
    /* right: 70px; */
}

.register::after {
    display: block;
    content: "";
    height: 2px;
    width: 106px;
    position: absolute;
    background-color: #f5f5f5;
    right: 70px;
    bottom: 3px;
}

.form-title::before {
    display: block;
    content: "";
    height: 49px;
    width: 2px;
    position: absolute;
    background-color: #f5f5f5;
    left: 50%;
    transform: translateX(calc(-50% + 10px));
}

.register:hover {
    opacity: 0.8;
}

```

```

.gr-input input {
    display: block;
    border: none;
    border-radius: 6px;
    -webkit-box-shadow: none;
    -moz-box-shadow: none;
    box-shadow: none;
    background-color: #f5f5f5;
    color: #1A374D;
    width: 100%;
    height: 60px;
    font-size: 19px;
    margin-bottom: 13px;
    padding-left: 20px;
    outline: 5px solid transparent;
    transition: all ease-in-out 0.35s;
}

.gr-input input:focus {
    outline-color: #04233c4f;
}

.gr-input {
    position: absolute;
    width: 80%;
    height: 310px;
    top: 110px;
    left: 50%;
    transform: translateX(-50%);
}

.input-email {}

.name {
    display: flex;
    width: 100%;
    justify-content: space-between;
}

```

Trên là những thuộc tính giúp cấu thành layout của form đăng ký.

Trong đó thuộc tính -webkit-box-shadow: none và -moz-box-shadow: none giúp hỗ trợ thuộc tính box-shadow trên hai trình duyệt Chrome và FireFox tốt hơn. Outline giúp tạo đường viền giữa border và padding. Transparent là màu trong suốt trong CSS.

## JavaScript của trang đăng nhập

```

// nút đăng ký
const btnRegister = document.querySelector("#btnRegister");
// thè thông báo
const warning = document.querySelector(".warning");
// input nhập họ và tên, email, password và nhập lại password
var firstName = document.getElementById("firstName");
var lastName = document.getElementById("lastName");
var email = document.getElementById("email");
var password = document.getElementById("password");
var rePassword = document.getElementById("rePassword");

// List user được lưu trong localstorage sẽ được lấy ra và chuyển sang kiểu mảng (trước đó trong local storage là kiểu chuỗi)
var listUser = localStorage.getItem('listUser');
if (listUser == null) {
    localStorage.setItem("listUser", JSON.stringify([]));
    listUser = localStorage.getItem('listUser');
}

// bắt sự kiện click cho nút đăng ký nếu người dùng chưa nhập đủ thông tin, hay là nhập mật khẩu ở hai ô nhập không giống nhau thì sẽ
// được thông báo rằng "Password đã nhập không trùng với bên trên"
// Nếu nhập hoàn tất và đúng yêu cầu thì sau khi nhập xong website sẽ thông báo rằng
// "Đăng ký thành công chuyển tới đăng nhập trong 4s nữa!" và sẽ tự động chuyển tới trang đăng nhập sau 4s
btnRegister.addEventListener("click", () => {
    if (email.value.length > 0 && firstName.value.length > 0 && lastName.value.length > 0 && password.value.length > 0 && rePassword.value.length > 0) {
        let tempFirstName = firstName.value;
        let tempLastName = lastName.value;
        let tempEmail = email.value;
        let tempPassword = password.value;
        var newUser;
        if (rePassword.value == password.value) {
            newUser = {
                email: tempEmail,
                firstName: tempFirstName,
                lastName: tempLastName,
                password: tempPassword
            }
        } else {
            warning.innerText = 'Password đã nhập không trùng với bên trên';
            warning.classList = "warning active";
            setTimeout(() => {
                warning.classList = "warning";
            }, 3000)
            return;
        }
    } else {
        warning.innerText = 'Password đã nhập không trùng với bên trên';
        warning.classList = "warning active";
        setTimeout(() => {
            warning.classList = "warning";
        }, 3000)
    }
    const parseList = JSON.parse(listUser);
    console.log(parseList);
    const newList = [...parseList, newUser];
    localStorage.setItem("listUser", JSON.stringify(newList));
    warning.innerText = 'Đăng ký thành công chuyển tới đăng nhập trong 4s nữa!';
    warning.style.backgroundColor = '#A0D995';
    warning.style.borderColor = "#6CC4A1";
    warning.classList = "warning active";
    setTimeout(() => {
        location.pathname = pathnameLoginPage;
    }, 4500);
} else {
    warning.innerText = 'Phải điền đầy đủ thông tin trong form!';
    warning.classList = "warning active";
    setTimeout(() => {
        warning.classList = "warning";
    }, 3000)
}
});

```

## Responsive của cả hai trang đăng ký và đăng

```

@media only screen and (max-width: 46.25em) {

    /* page login and register */
    .form {
        width: 97%;
    }

    .login-res.form {
        height: 400px;
    }

    .register-res.form {
        height: 475px;
    }

    .login, .register {
        font-size: 17px !important;
    }

    .login {
        left: 10px;
    }

    .register {
        right: 10px;
    }

    .form-title span {
        position: relative;
    }

    .form-title::before {
        transform: translateX(calc(-50% + 6px));
    }

    .register::after {
        width: 72px;
        bottom: -2px;
        right: 0;
    }

    .login::after {
        width: 98px;
        bottom: -2px;
        left: 0;
    }

    #btnLogin {
        bottom: 54px;
        width: 138px;
        height: 48px;
        font-size: 20px;
    }

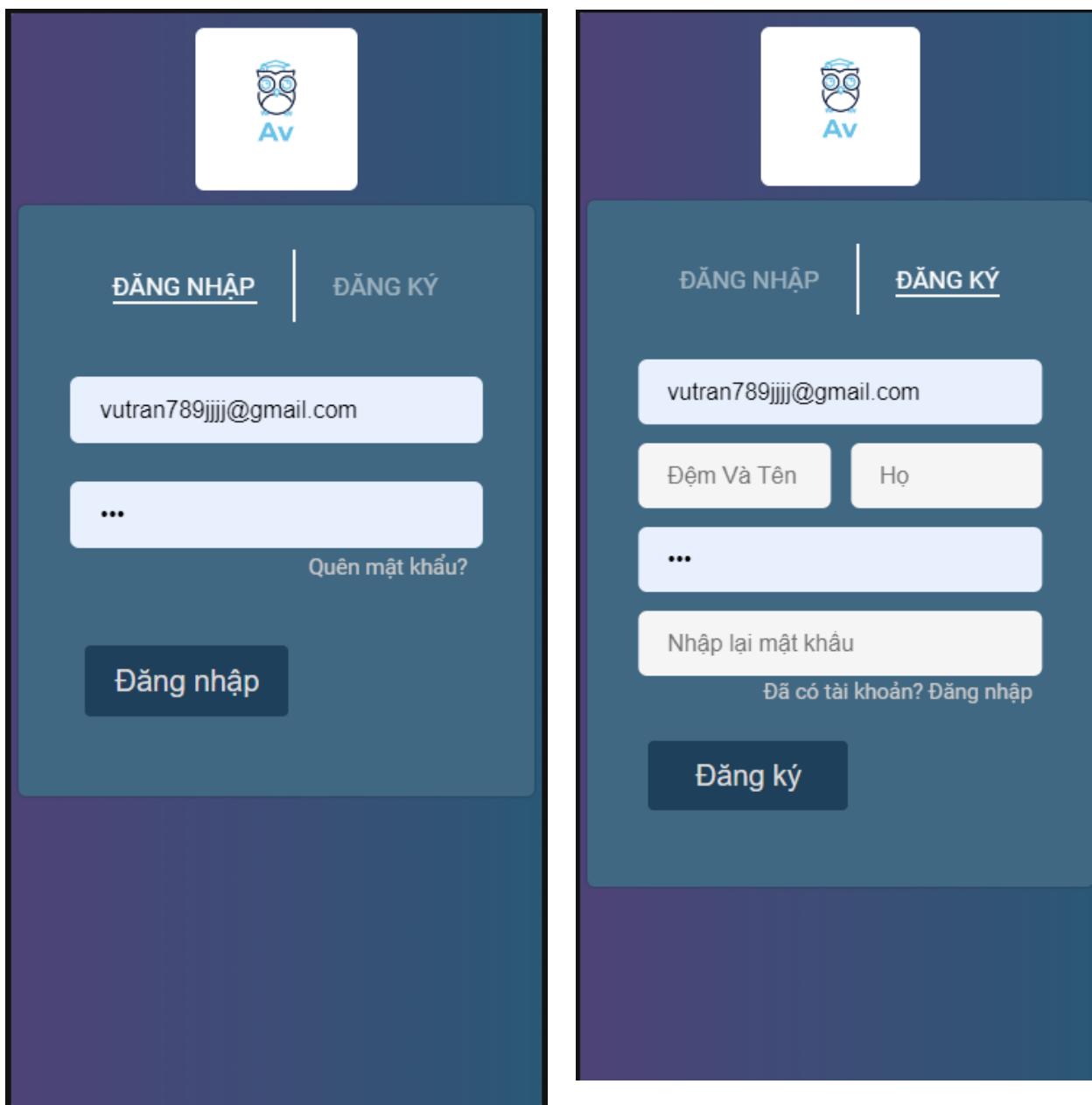
    .register-btn {
        width: 138px;
        height: 48px;
        font-size: 20px;
        left: 42px;
        bottom: 53px;
    }

    .forgot-password {
        right: 45px;
        bottom: 147px;
        font-size: 15px;
    }

    .have-account {
        right: 42px;
        bottom: 127px;
        font-size: 15px;
    }

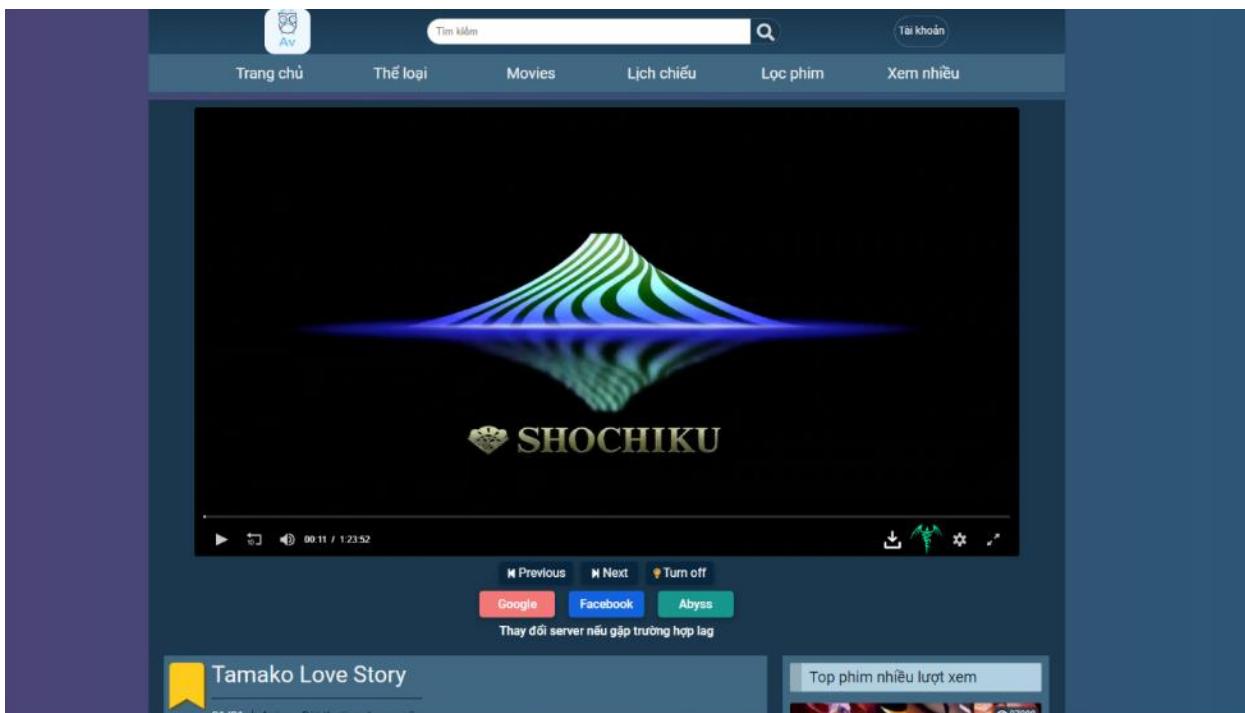
    .input input, .gr-input input {
        height: 45px;
    }
}

```

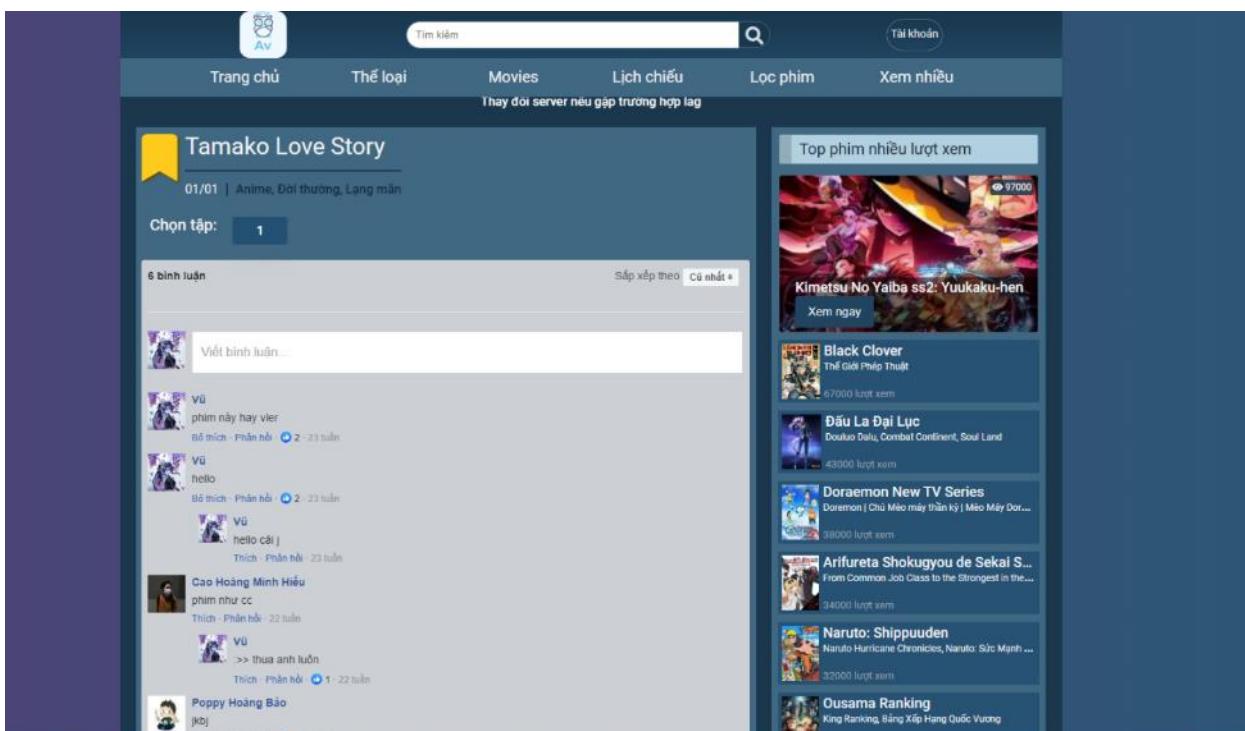


Giao diện của trang đăng ký và đăng nhập sẽ tự co nhỏ lại kể từ 46.25em (729.6px). Ngoài việc co nhỏ về giao diện thì font-size cũng sẽ nhỏ lại để thích hợp hơn so với màn hình nhỏ như trên các thiết bị mobile.

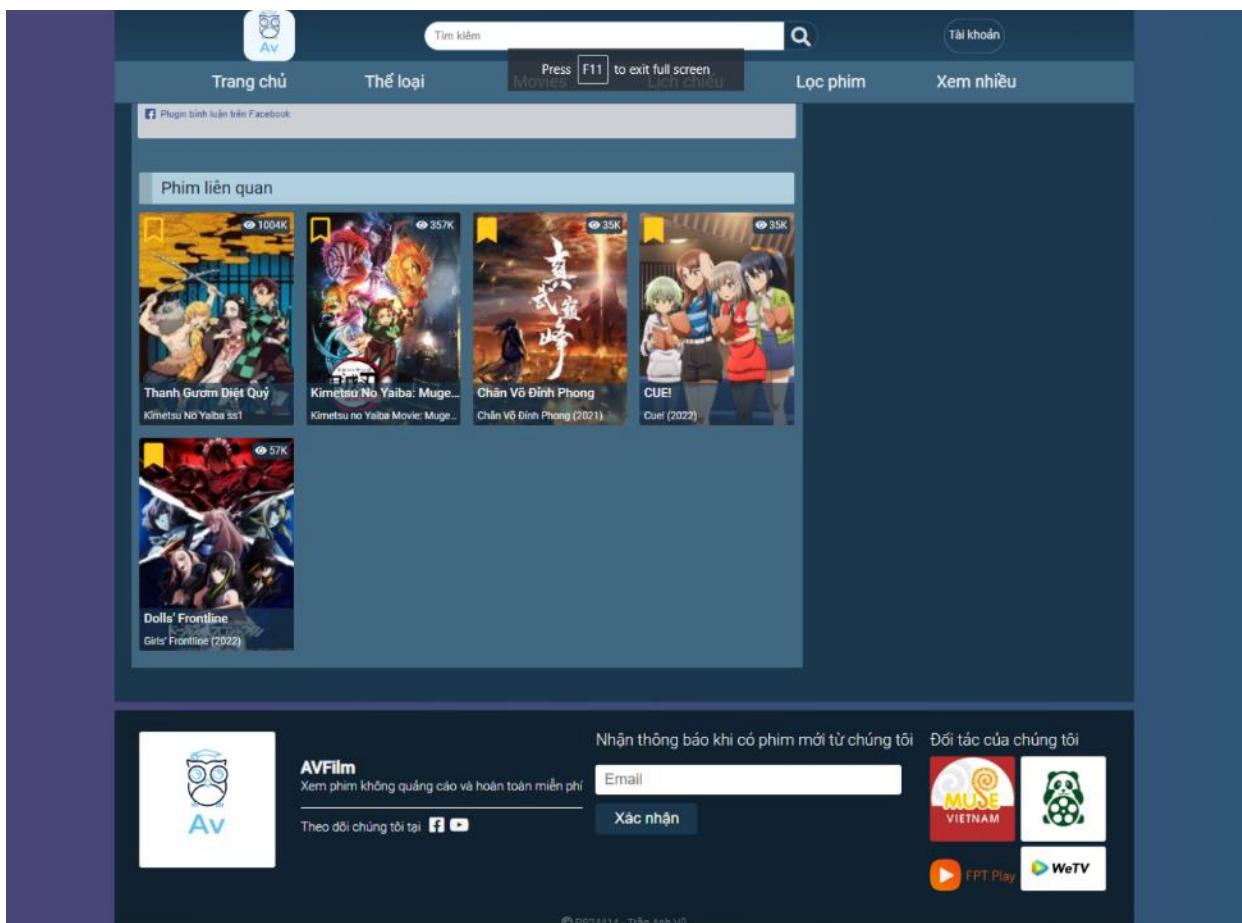
- Trang xem phim



The screenshot shows a movie player interface. At the top, there's a navigation bar with links: Trang chủ, Thể loại, Movies, Lịch chiếu, Lọc phim, and Xem nhiều. Below the navigation bar is a large video frame displaying a dark background with a green and blue abstract pattern, and the word "SHOCHIKU" at the bottom. A control bar at the bottom of the video frame includes buttons for play, pause, volume, and time (00:11 / 1:23:52). Below the video frame are several control buttons: Previous, Next, Turn off, Google, Facebook, and Abyss. A message "Thay đổi server nếu gặp trường hợp lag" is displayed. At the bottom left, there's a thumbnail for "Tamako Love Story" with the text "01/01 | Anime, Đời thường, Lãng mạn". On the right side, there's a "Top phim nhiều lượt xem" section listing several anime titles with their cover art and view counts.

This screenshot shows the same movie player interface, but the main content area now displays the episode list for "Tamako Love Story". The list shows 6 bình luận (comments) from users like Vũ, Cao Hoàng Minh Hiếu, and Poppy Hoàng Bảo. To the right of the episode list is the same "Top phim nhiều lượt xem" section as in the previous screenshot.



Header và Footer của Trang xem phim sẽ không có sự thay đổi sao với trang chi tiết phim. Trang xem phim gồm thành phần chính là phim được chiếu ở đầu trang phần chọn tập phim.

```
<div class="video-film-container">
  <div class="box" id="loader">
    <div class="cat">
      <div class="cat__body"></div>
      <div class="cat__body"></div>
      <div class="cat__tail"></div>
      <div class="cat__head"></div>
    </div>
  </div>
</div>
```

```
.video-film-container {
  width: 100%;
  height: 100%;
  background-color: #072842;
  border-radius: 5px;
  box-shadow: inset 0 0 10px #021b2f;
}
```

“video-film-container” là thẻ được dùng để chứa phim với kích thước chiều ngang là 100% so với thẻ cha chứa nó là “watch-container” và chiều rộng là 100% so với thẻ cha chứa nó. Thuộc tính box-shadow: inset 0 0 10px #021b2f; giúp cho thẻ sẽ có bóng đổ vào bên trong.

```
.watch-container {  
    --heightOfVideo: calc(var(--widthOfVideo) * 9 / 16);  
    max-width: 1080px;  
    width: var(--widthOfVideo);  
    height: var(--heightOfVideo);  
    margin: 0 auto;  
    padding: 10px 0;  
    z-index: 3;  
    position: relative;  
}
```

Thẻ “watch-container” sẽ có chiều rộng phụ thuộc vào biến –widthOfVideo

Và chiều cao sẽ phụ thuộc vào biến –heightOfVideo. Biến –widthOfVideo sẽ được tính toán bởi javascript khi người dùng thay đổi kích thước trình duyệt từ đó biến –heightOfVideo cũng được tính toán lại và sẽ giữ đảm bảo tỉ lệ 16:9 để video không bị bể. Z-index: 3 giúp cho thẻ có thể nổi lên trên so với thẻ có thuộc tính position mà có z-index nhỏ hơn 3.

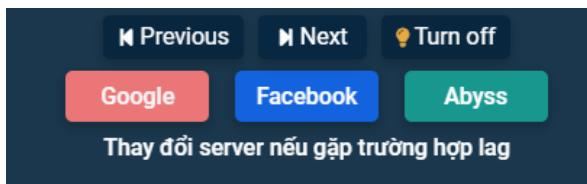
```
resizeWatchContainer();  
window.onresize = () => {  
    resizeWatchContainer();  
}  
  
function resizeWatchContainer() {  
    if (window.innerWidth <= 1080) {  
        console.log(true);  
        watchContainer.setAttribute("style", `--widthOfVideo: ${window.innerWidth - 5}px`);  
    } else {  
        watchContainer.setAttribute("style", `--widthOfVideo: 1080px`)  
    }  
}
```

Code JavaScript giúp tính lại chiều rộng cho “watch-container”. Trong đó window.innerWidth giúp lấy ra được chiều rộng của trình duyệt hiện tại. “watchContainer” là biến của thẻ “watch-container”, setAttribute(“style”, value);

Giúp xét thuộc tính style inline của thẻ “watch-container” theo giá trị được tính.

```
<div class="navigation_video">
  <div class="previous_episode" id="btnPrevious">
    <i class="fas fa-step-backward"></i> Previous
  </div>
  <div class="next_episode" id="btnNext">
    <i class="fas fa-step-forward"></i> Next
  </div>
  <div class="toggle_light" id="btnToggleLight">
    <i class="fas fa-lightbulb"></i> Turn off
  </div>
  <div class="mark"></div>
</div>
```

```
.navigation_video {
  display: flex;
  width: 100%;
  gap: 10px;
  height: 45px;
  justify-content: center;
  align-items: center;
}
```



```
<div class="server-container">
  <span class="server-item google">Google</span>
  <span class="server-item facebook">Facebook</span>
  <span class="server-item abyss">Abyss</span>
</div>
<p class="note-server">Thay đổi server nếu gặp trường hợp lag</p>
```

“navigation\_video” là thẻ chứa các phím điều hướng như next, previous, hay tắt đèn. Còn “server-container” chứa các phím chức năng để đổi server khi người dùng bị lag trong quá trình xem phim. “navigation\_video” sẽ có display: flex giúp dàn layout cho những thẻ con trong nó theo chiều ngang và chiều rộng của thẻ là 100% thẻ cha chứa nó.

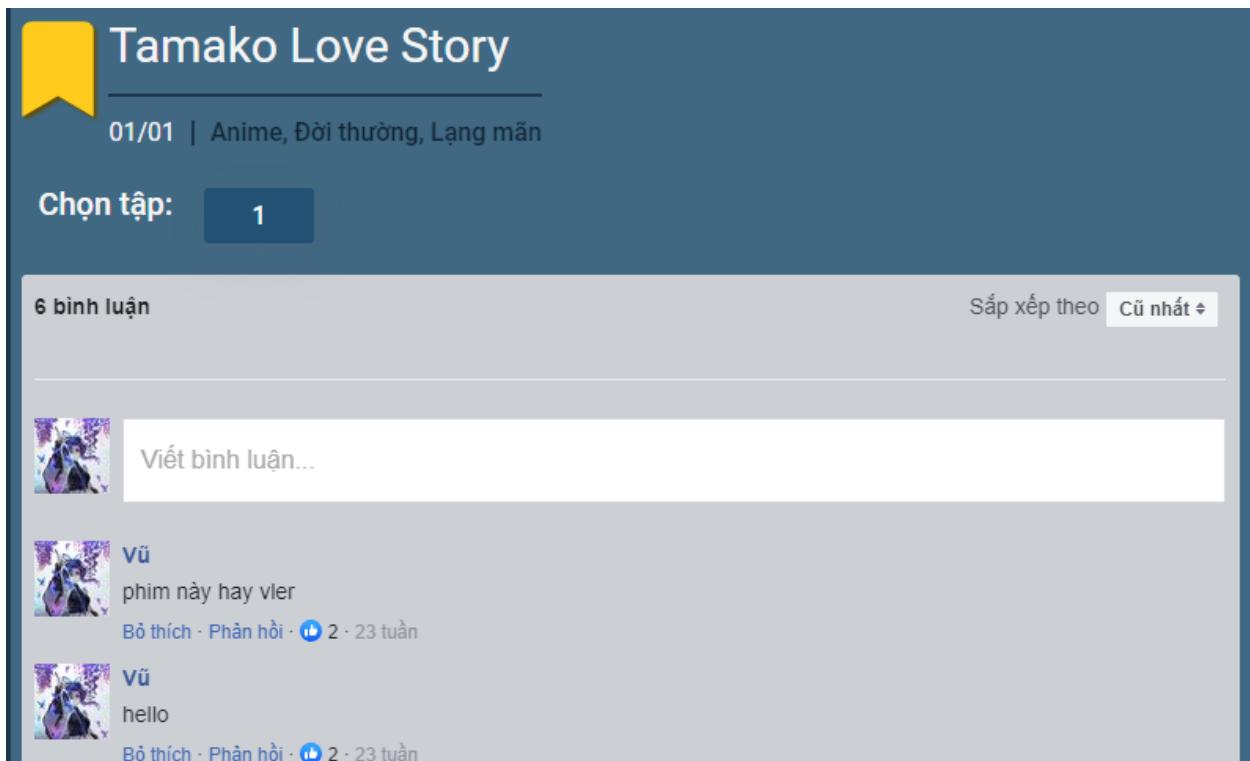
Code JavaScript của các nút điều hướng

```
btnNext.addEventListener("click", () => {
  console.log(currentEpisodeIndex);
  console.log(arrEpisode.length);
  if (currentEpisodeIndex >= arrEpisode.length - 1) return;
  setCurrentSlugVideo(arrEpisode[currentEpisodeIndex + 1].getAttribute("data-slug"));
})
btnPrevious.addEventListener("click", () => {
  console.log(currentEpisodeIndex);
  console.log(arrEpisode.length);
  if (currentEpisodeIndex == 0) return;
  setCurrentSlugVideo(arrEpisode[currentEpisodeIndex - 1].getAttribute("data-slug"));
})

btnToggleLight.addEventListener("click", () => {
  console.log(mark);
  mark.classList.toggle("turn_on");
  if (mark.classList.contains("turn_on")) {
    btnToggleLight.innerHTML = `<i class="fas fa-lightbulb" style="color: white;"></i> Turn on`;
  } else {
    btnToggleLight.innerHTML = `<i class="fas fa-lightbulb" style="color: #E8AA42;"></i> Turn off`;
  }
})
```

```
export function loadVideoFilm(container, currentSlug) {
    if (currentSlug === undefined) {
        var temp = `Hiện website chưa cập nhật bộ phim này. Xin lỗi bạn vì sự bất tiện`;
        container.innerHTML += temp;
    } else {
        container.innerHTML =
            `
```

Code JavaScript giúp tải phim lên website với đối số “container” là thẻ chứa phim và “currentSlug” giúp xác định để chiếu đúng tập phim mà người dùng chọn.



The screenshot shows a video player interface for the anime 'Tamako Love Story'. At the top, there's a yellow bookmark icon and the title 'Tamako Love Story'. Below the title, it says '01/01 | Anime, Đời thường, Lãng mạn'. A button labeled 'Chọn tập:' has '1' selected. On the left, there's a comment section header '6 bình luận'. On the right, there are two comments from a user named 'Vũ': one saying 'phim này hay vler' and another saying 'hello'. Both comments have 'Bỏ thích · Phản hồi · 2 · 23 tuần' below them.

```
<div class="film-detail">
    <div class="film-title-pr">
        <div class="film-detail-bookmark">
            <i class="${bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon-detail"></i>
        </div>
        <div class="title-pr">
            <div class="name-film-pr">
                ${filmName}
            </div>
            <div class="film-category__epsoide">
                <p class="film-epsoide-pr">${episode}</p>
                <div class="film-category-pr">${type}</div>
            </div>
        </div>
    </div>
    <div class="choice_episode_field">
        <div class="choice_episode-title">Chọn tập: </div>
        <div id="episode-container">
        </div>
    </div>
    <div class="comment-field">
        <div class="fb-comments"
            data-href="https://travux.github.io/AVFilm-demo/film-detail.html" data-width="100%"
            data-numposts="10">
        </div>
    </div>
</div>
```

Thẻ “film-detail” phần lớn sẽ kế thừa giao diện của phần chi tiết phim trong trang Chi tiết phim. Điểm khác biệt là sẽ có thẻ thẻ “choice\_episode-field” để có thể chứa những tập phim của bộ phim đó.

Code JavaScript giúp load tập phim

```
let episodeContainer = document.querySelector("#episode-container");
slugs.map((slug, index) => {
    episodeContainer.innerHTML += `<div class="a-episode" data-slug="${slug}">${index + 1}</div>`;
});
// here set the event onClick for an episode container
let episodeArr = document.querySelectorAll(".a-episode");
if (episodeArr != null) {
    episodeArr.forEach(episode => {
        episode.addEventListener("click", () => {
            setCurrentSlugVideo(episode.getAttribute("data-slug"));
        });
    });
}
```

“episodeContainer” là biến đại diện cho thẻ chứa các tập phim. “slugs” là mảng những chuỗi ký tự giúp xác định đúng tập phim của bộ phim đó. Vòng lặp sẽ chạy để có thể lặp qua tất cả các phần tử trong mảng mỗi khi qua một phần tử nó sẽ thêm vào “episodeContainer” 1 thẻ đại diện cho 1 tập phim với nội dung hiện thị sẽ là số thứ tự của tập phim

“episodeArr” là mảng các thẻ đại diện của mỗi tập phim. Nếu mảng này khác null thì mỗi phần tử trong mảng sẽ được gán cho lắng nghe sự kiện ‘click’. Giúp có thể lưu vào Local storage đúng slug của tập phim hiện tại và tải lại trang để có thể load tập phim đó lên.

- **Những hàm JavaScript dùng nhiều trong website**

```
export function loadTopView(container) {
    container.innerHTML += `
        <div class="film-top-header" id="filmTopViewHeader">Top phim nhiều lượt xem</div>
    `;
    topView.map((film) => {
        var temp;
        if (film.top1) {
            temp = `
                <div class="film-top1-container top1view">
                    
                    <span class="film-top1-name">
                        ${film.filmName}
                    </span>
                    <div class="film-view">
                        <i class="fas fa-eye"></i> ${film.view}
                    </div>
                </div>
                <div class="film-top-container topview mobile-respon">
                    
                    <div class="film-top-title">
                        <span class="name">${film.filmName}</span>
                        <span class="sub"><br> ${film.subTitle}</span>
                    </span>
                    <span class="view">${film.view} lượt xem</span>
                </div>
            `;

        } else {
            temp = `
                <div class="film-top-container topview">
                    
                    <div class="film-top-title">
                        <span class="name">${film.filmName}</span>
                        <span class="sub"><br> ${film.subTitle}</span>
                    </span>
                    <span class="view">${film.view} lượt xem</span>
                </div>
            `;
        }
        container.innerHTML += temp;
    });
}
```

**“loadTopView”** giúp load lên những phim đang ở trên top lượt xem, những trang trong website sử dụng hàm này là **Trang chủ**, **Trang chi tiết phim**, **Trang xem phim**.

```

export function loadTopComment(container) {
    container.innerHTML += `
        <div class="film-top-comment-header film-top-header" id="filmTopViewHeader">Top phim nhiều bình luận</div>
    `;
    topComment.map(film => {
        var temp;
        if (film.top1) {
            temp = `
                <div class="film-top1-container top1cmt">
                    
                    <span class="film-top1-name">
                        ${film.filmName} <br>
                        <button class="watch-btn btn">Xem ngay</button>
                    </span>
                    <div class="film-cmt">
                        <i class="far fa-comment"></i> ${film.comment}
                    </div>
                </div>
                <div class="film-top-container topcmt mobile-respon">
                    
                    <div class="film-top-title">
                        <span class="name">${film.filmName}</span>
                        <span class="sub"><br>${film.subTitle}</span>
                    </span>
                    <span class="view">${film.comment} lượt bình luận</span>
                </div>
            `;
        } else {
            temp = `
                <div class="film-top-container topcmt">
                    
                    <div class="film-top-title">
                        <span class="name">${film.filmName}</span>
                        <span class="sub"><br>${film.subTitle}</span>
                    </span>
                    <span class="view">${film.comment} lượt bình luận</span>
                </div>
            `;
        }
        container.innerHTML += temp;
    });
}

```

**“loadTopComment”** giúp load lên những phim đang ở trên top lượt bình luận, những trang trong website sử dụng hàm này là **Trang chủ**, **Trang chi tiết phim**.

```
// export function setOnClickForFilmTopCmt() {
//     const filmTop1Cmt = document.querySelector('.top1cmt');
//     const filmTopCmt = document.querySelectorAll('.topcmt');
//     filmTop1Cmt.addEventListener("click", () => {
//         sessionStorage.setItem('indexFilm', 0);
//         sessionStorage.setItem('typeOfData', 'topCmt');
//         location.pathname = pathNameDetailPage;
//     });
//     filmTopCmt.forEach((element, index) => {
//         element.addEventListener("click", () => {
//             sessionStorage.setItem('indexFilm', index);
//             sessionStorage.setItem('typeOfData', 'topCmt');
//             location.pathname = pathNameDetailPage;
//             console.log(2);
//         });
//     });
// }

// export function setOnClickForFilmTopView() {
//     const filmTop1View = document.querySelector('.top1view');
//     const filmTopView = document.querySelectorAll('.topview');
//     filmTop1View.addEventListener("click", () => {
//         sessionStorage.setItem('indexFilm', 0);
//         sessionStorage.setItem('typeOfData', 'topView');
//         location.pathname = pathNameDetailPage;
//     });
//     filmTopView.forEach((elementFilm, index) => {
//         elementFilm.addEventListener("click", () => {
//             sessionStorage.setItem('indexFilm', index);
//             sessionStorage.setItem('typeOfData', 'topView');
//             location.pathname = pathNameDetailPage;
//         });
//     });
// }
```

**“setOnClickForFilmTopCmt”** và **“setOnClickForFilmTopView”** giúp thêm sự kiện click cho mỗi bộ phim của top lượt xem và top bình luận. Những trang trong website sử dụng hàm này là **Trang chủ**, **Trang chi tiết phim**, **Trang xem phim**.

```

export function loadFilmDetail(mainContainer, beforeElement, { thumbnail, filmName, subTitle, urlVideo, desc, bookmark, episode, backgroundImg, type, slugs }) {
    if (localStorage.currentSlug != null) {
        localStorage.setItem("currentSlug", slugs[0]);
        console.log(slugs);
    }
    var filmElement = document.createElement('div');
    var temp = '';
    <div class="film-detail">
        <div class="film-title-pr">
            <div class="film-detail-bookmark">
                <i class="${bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon-detail"></i>
            </div>
            <div class="title-pr">
                <div class="name-film-pr">
                    ${filmName}
                </div>
                <div class="film-category_epsoide">
                    <p class="film-episode-pr">${episode}</p>
                    <div class="film-category-pr">${type}</div>
                </div>
            </div>
        </div>
        <div class="film-img-pr" style="background-image: url('${backgroundImg}');">
            <div class="thumbnail">
                
            </div>
            <div class="dec">
                <p class="title-dec">Nội dung phim</p>
                <a href="">
                    <p class="name-film-dec">${filmName}</p>
                </a>
                <p class="article-dec">${desc}</p>
                <button class="trailer"><a target="_blank" rel="noopener noreferrer" href="${urlVideo}">Xem trailer</a></button>
                <button class="choice-episode" id="watchFilm">Xem phim</button>
            </div>
        </div>
        <div class="comment-field">
            <div class="fb-comments" data-href="https://travux.github.io/AVFilm-demo/film-detail.html"
                data-width="100%" data-numposts="10">
            </div>
        </div>
    </div>
    filmElement.innerHTML += temp;
    mainContainer.insertBefore(filmElement, beforeElement);
}

```

**"loadFilmDetail"** giúp load ra chi tiết mà bộ phim người dùng đã bấm vào, nó được sử dụng ở một số trang trong website như **Trang chủ, Trang lịch chiếu, Trang thể loại phim, Trang quản lý tài khoản, Trang chi tiết phim, Trang xem phim.**

```

export function loadFilmWatchDetail(mainContainer, beforeElement, { filmName, bookmark, episode, type, slugs }) {
    if (localStorage.currentSlug == null) {
        localStorage.setItem("currentSlug", slugs[0]);
    }
    var filmElement = document.createElement('div');
    var temp = `

        <div class="film-detail">
            <div class="film-title-pr">
                <div class="film-detail-bookmark">
                    <i class="${bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon-detail"></i>
                </div>
                <div class="title-pr">
                    <div class="name-film-pr">
                        ${filmName}
                    </div>
                    <div class="film-category__epsoide">
                        <p class="film-epsoide-pr">${episode}</p>
                        <div class="film-category-pr">${type}</div>
                    </div>
                </div>
            </div>
            <div class="choice_episode_field">
                <div class="choice_episode-title">Chọn tập: </div>
                <div id="episode-container">
                </div>
            </div>
            <div class="comment-field">
                <div class="fb-comments"
                    data-href="https://travux.github.io/AVFilm-demo/film-detail.html" data-width="100%"
                    data-numposts="10">
                </div>
            </div>
        </div>
    `;
    filmElement.innerHTML += temp;
    mainContainer.insertBefore(filmElement, beforeElement);
    // here is fill amount of episodes into the container
    let episodeContainer = document.querySelector("#episode-container");
    slugs.map((slug, index) => {
        episodeContainer.innerHTML += `<div class="a-episode" data-slug="${slug}">${index + 1}</div>`;
    });
    let episodeArr = document.querySelectorAll(".a-episode");
    if (episodeArr != null) {
        episodeArr.forEach(episode => {
            episode.addEventListener("click", () => {
                setCurrentSlugVideo(episode.getAttribute("data-slug"));
            });
        });
    }
}

```

"**loadFilmWatchDetail**" giúp load ra chi tiết của bộ phim và cả số tập phim của bộ phim đó cho người dùng chọn để xem, trang sử dụng hàm này là **Trang xem phim**

```

export function setCurrentSlugVideo(slug) {
    localStorage.setItem("currentSlug", slug);
    location.pathname = pathnameWatchPage;
}

export function loadVideoFilm(container, currentSlug) {
    if (currentSlug === undefined) {
        var temp = `Hiện website chưa cập nhật bộ phim này. Xin lỗi bạn vì sự bất tiện`;
        container.innerHTML += temp;
    } else {
        container.innerHTML = `
            <iframe width="1080" height="607.5" src="https://short.link/${currentSlug}" frameborder="0" scrolling="0"
            allowfullscreen></iframe>
        `;
    }
}

```

**"setCurrentSlugVideo"** giúp thay đổi giá trị slug của video hiện tại trong local storage của website. Hàm này sẽ được dùng trong hàm "**loadFilmDetail**" và hàm "**loadFilmWatchDetail**".

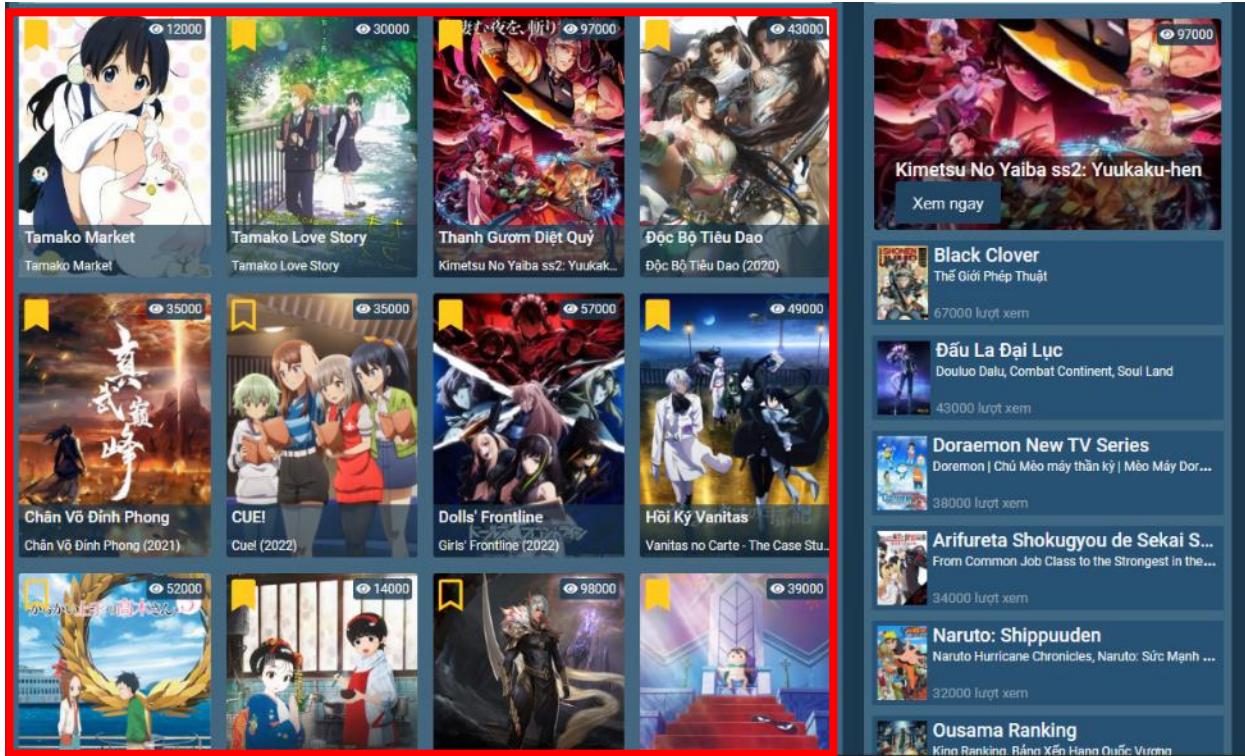
**"loadVideoFilm"** giúp tải lên tập phim mà người dùng đã bấm vào trước đó, trang sử dụng hàm này là **Trang xem phim**.

```

export function loadFilm(container, listFilm, typeOfData, className) {
    listFilm.map((film) => {
        var temp = `
            <div class="film-container ${className}">
                <div class="film-view">
                    <i class="fas fa-eye"></i> ${film.view}
                </div>
                <div class="film-bookmark" id="film-bookmark">
                    <i class=" ${film.bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon"></i>
                </div>
                
                <div class="film-title">
                    <p class="main-title">${film.filmName}</p>
                    <p class="sub">${film.subTitle}</p>
                </div>
                <div class="mask">
                    <i class="far fa-play-circle"></i>
                    <p class="info">Mô tả</p>
                    <p class="episode">${film.episode}</p>
                </div>
            </div>
        container.innerHTML += temp;
    });
    const aFilmContainer = document.querySelectorAll(`.film-container.${className}`);
    aFilmContainer.forEach((element, index) => {
        element.addEventListener("click", () => {
            sessionStorage.setItem('indexFilm', index);
            sessionStorage.setItem('typeOfData', typeOfData);
            location.pathname = pathnameDetailPage;
        })
    });
}

```

"loadFilm" giúp tải ra những bộ phim vào phần chính của website giúp người dùng có thể chọn phim.



Những trang sử dụng hàm này trong website là **Trang chủ**, **Trang thể loại**.

```

export function loadFilmCategory(container, category) {
    const filmCategoryHeader = document.getElementById('filmCategoryHeader');
    filmCategoryHeader.innerText = sessionStorage.getItem('category');
    var arrIndex = [];
    filmsInMain.map((film, index) => {
        var temp;
        if (film.type.includes(category)) {
            temp = `
                <div class="film-container">
                    <div class="film-view">
                        <i class="fas fa-eye"></i> ${film.view}
                    </div>
                    <div class="film-bookmark" id="film-bookmark">
                        <i class=" ${film.bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon"></i>
                    </div>
                    
                    <div class="film-title">
                        <p class="main-title">${film.filmName}</p>
                        <p class="sub">${film.subTitle}</p>
                    </div>
                    <div class="mask">
                        <i class="far fa-play-circle"></i>
                        <p class="info">Mô tả</p>
                        <p class="episode">${film.episode}</p>
                    </div>
                </div>
            `;
            container.innerHTML += temp;
            arrIndex.push(index);
        }
    });
    const aFilmContainer = document.querySelectorAll('.film-container');
    aFilmContainer.forEach((element, index) => {
        element.addEventListener("click", () => {
            sessionStorage.setItem('indexFilm', arrIndex[index]);
            sessionStorage.setItem('typeOfData', 'detail')
            location.pathname = pathnameDetailPage;
        })
    });
}

```

**“loadFilmCategory”** giúp tải lên những bộ phim theo thể loại. Trang sử dụng hàm này là **Trang thể loại**.

```

export function setCategory(arrItemCategory) {
    arrItemCategory.forEach(item => {
        item.addEventListener('click', () => {
            sessionStorage.setItem('category', item.innerText);
        });
    });
}

```

“**setCategory**” giúp thêm sự kiện click cho những thẻ chọn thể loại phim để có thể thay đổi biến “**category**” trong Local Storage.



Những trang sử dụng hàm này là **Trang chủ**, **Trang chi tiết phim**, **Trang xem phim**, **Trang thể loại**, **Trang lịch chiếu**

```

export function loadFilmShowTime(container) {
    showTimes.map((film, index) => {
        var temp = `
            <div class="film-container" name="${index}">
                <div class="film-episode">Tập ${film.newEpisode}</div>
                <div class="film-view">
                    <i class="fas fa-eye"></i> ${film.view}
                </div>
                <div class="film-bookmark" id="film-bookmark">
                    <i class=" ${film.bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon"></i>
                </div>
                
                <div class="film-title">
                    <p class="main-title">${film.filmName}</p>
                    <p class="sub">${film.subTitle}</p>
                </div>
                <div class="mask">
                    <i class="far fa-play-circle"></i>
                    <p class="info">Mô tả</p>
                    <p class="episode">${film.episode}</p>
                </div>
            </div>
        `;
        switch (film.date) {
            case 'monday':
                container[0].innerHTML += temp;
                break;
            case 'tuesday':
                container[1].innerHTML += temp;
                break;
            case 'wednesday':
                container[2].innerHTML += temp;
                break;
            case 'thursday':
                container[3].innerHTML += temp;
                break;
            case 'friday':
                container[4].innerHTML += temp;
                break;
            case 'saturday':
                container[5].innerHTML += temp;
                break;
            case 'sunday':
                container[6].innerHTML += temp;
                break;
            default:
                throw new Error('Invalid date of showtimes');
        }
    });
}

export function setOnClickForFilmShowTime() {
    const arrFilmShowTime = document.querySelectorAll(".film-container");
    arrFilmShowTime.forEach(film => {
        film.addEventListener('click', () => {
            const indexFilm = film.getAttribute("name");
            sessionStorage.setItem('indexFilm', indexFilm);
            sessionStorage.setItem('typeOfData', 'showtime')
            location.pathname = pathnameDetailPage;
        });
    })
}
}

```

**"loadFilmShowTime"** giúp tải lên những bộ phim theo lịch (gồm các ngày trong tuần), trang sử dụng hàm này là **Trang lịch chiếu**

**"setOnClickForFilmShowTime"** giúp thêm sự kiện click cho những bộ phim trong trang **Lịch chiếu**

```

export function changeUserContainer(userContainer, mobileUserContainer, indexUser) {
    userContainer.innerHTML = '';
    mobileUserContainer.innerHTML = "";
    if (indexUser != -1) {
        var currentUser = JSON.parse(localStorage.getItem("listUser"))[indexUser];
        var userName = currentUser.firstName;
        userContainer.innerHTML += `
            
            <span class="user-name">${userName}</span>
            <ul class="user-list">
                <li class="list-item">
                    <a href="#"><i class="fas fa-user item-icon"></i></a>
                    <a href="user.html">Quản lý tài khoản</a>
                </li>
                <li class="list-item">
                    <a href="#"><i class="fas fa-bookmark item-icon"></i></a>
                    <a href="#">Phim đã lưu</a>
                </li>
                <li class="list-item">
                    <a href="#"><i class="fas fa-sign-out-alt item-icon"></i></a>
                    <a href="#" id="btnLogout">Đăng xuất</a>
                </li>
            </ul>
        `;
        mobileUserContainer.innerHTML += `
            
            <div class="user-name-mobile">${userName}</div>
        `;
    } else {
        userContainer.innerHTML += `
            <span class="user-name none-login">Tài khoản</span>
            <ul class="user-list list-none-login">
                <li class="list-item">
                    <a href="login.html"><i class="fas fa-sign-in-alt item-icon"></i></a>
                    <a href="login.html">Đăng nhập</a>
                </li>
                <li class="list-item">
                    <a href="register.html"><i class="fas fa-user-plus item-icon"></i></a>
                    <a href="register.html">Đăng ký</a>
                </li>
            </ul>
        `;
        mobileUserContainer.innerHTML += `
            <div class="user-name-mobile" id="btnUser">Tài khoản</div>
        `;
    }
}

```

**“changeUserContainer”** giúp thay đổi trang thái cũng như nội dung của thẻ **“user-container”** đã đăng nhập hay chưa đăng nhập cho website.



Nếu đã đăng nhập thì thẻ này sẽ hiển thị Tên kèm ảnh đại diện và ngược lại sẽ hiển thị Tài Khoản nếu chưa đăng nhập. Những trang trong website sử dụng hàm trên là **Trang chủ**, **Trang chi tiết phim**, **Trang lịch chiếu**, **Trang thể loại**, **Trang xem phim**, **Trang quản lý tài khoản**.

```
export function loadFlmBookmark(container) {
    var count = 0;
    filmsInMain.map(film => {
        var temp;
        if (film.bookmark) {
            var temp =
                <div class="film-container">
                    <div class="film-view">
                        <i class="fas fa-eye"></i> ${film.view}
                    </div>
                    <div class="film-bookmark" id="film-bookmark">
                        <i class=" ${film.bookmark ? 'fas' : 'far'} fa-bookmark bookmark-icon"></i>
                    </div>
                    
                    <div class="film-title">
                        <p class="main-title">${film.filmName}</p>
                        <p class="sub">${film.subTitle}</p>
                    </div>
                    <div class="mask">
                        <i class="far fa-play-circle"></i>
                        <p class="info">Mô tả</p>
                        <p class="episode">${film.episode}</p>
                    </div>
                </div>
                container.innerHTML += temp;
                count++;
        }
    });
    const aFilmContainer = document.querySelectorAll('.film-container');
    aFilmContainer.forEach((element, index) => {
        element.addEventListener("click", () => {
            sessionStorage.setItem('indexFilm', index);
            sessionStorage.setItem('typeOfData', 'detail');
            location.pathname = pathnameDetailPage;
        })
    });
}
```

“**loadFilmBookmark**” giúp tải lên những bộ phim đã được đánh dấu vàng ở góc trên bên trái vào trong phần **Phim đã lưu** của trang **Quản lý tài khoản**.

```
export function slideHeaderCenter(headerCenter, filmNameHeaderCenter, filmSubHeaderCenter, btnHeaderCenter) {
    var arrImage = [];
    for (let i = 0; i < arrBackground.length; i++) {
        let img = new Image();
        img.src = arrBackground[i].filmBackgroud;
        arrImage.push(img);
    }
    var index = 0;
    setInterval(() => {
        if (index >= arrImage.length) {
            index = 0;
        }
        headerCenter.style.backgroundImage = `url(${arrImage[index].src})`;
        filmNameHeaderCenter.innerHTML = arrBackground[index].filmName;
        filmSubHeaderCenter.innerHTML = arrBackground[index].filmSub;
        btnHeaderCenter.setAttribute("data-indexfilm", arrBackground[index].indexInMain);
        index++;
    }, 5000);
}
```

“**slideHeaderCenter**” giúp thay đổi ảnh banner cũng như nội dung khi hover vào của ảnh, những trang trong website sử dụng hàm này là **Trang chủ**, **Trang Thể loại**, **Trang lịch chiếu**.

```
scripts > data.js > showTimes
> export const filmsInMain = [ ... ];
> export const topView = [ ... ];
> export const topComment = [ ... ];
> export const filmsTrend = [ ... ];
> export const showTimes = [ ... ];
> export const arrBackground = [ ... ];
```

Đây là những mảng Object được lưu trữ trong file **data.js**. “**filmsInMain**” là những bộ phim có trong trang chủ, “**topView**” là những bộ phim nằm ở trong top nhiều lượt xem, “**topComment**” là những bộ phim nằm ở trong top nhiều lượt bình

luận, “**filmsTrend**” là những bộ phim được xem là top thịnh hành, “**showTimes**” là những bộ phim có tập phim sẽ chiếu sắp tới, “**arrBackGround**” là những bộ phim xuất hiện trên banner.