Functionality

## {Giang} [API] Watch Movie

<https://localhost:5001/api/Video/GetVideoContent?videoId=1>

* Stream video qua HTTP = API:
  + Asynchronous Video Streaming: gửi từng packets dữ liệu đến client thay vì gửi cả 1 file video => Dùng class FileStream
  + => Giảm áp lực phía server

Ng dùng nhận dữ liệu nhanh hơn

* Client: *HTML5 <video>*
  + Source: API URL
* Server:
  + Client cứ request data (tua/xem) -> server gửi data
  + Client disconnect -> kết thúc stream

## Database Design

* **Database**:
  + **User**
    - UserId: int
    - Email: string
    - Username: string
    - Password: string
    - RoleId: int
  + **Role**
    - RoleId: int
    - RoleName: string
  + **Movie**
    - MovieId: int
    - MovieName: string
    - VideoPath: string (đường dẫn đến video của phim, phía API)
    - Description: string
    - Duration: int (thời lượng phim – tính theo phút)
    - Rated: int (độ tuổi xem)
    - PublishedYear: int
    - Country: string
    - ImagePath: string (đường dẫn đến ảnh poster của phim, phía Client)
  + **Category**
    - ConnectionId: int (thể hiện kết nối giữa 2 table)
    - CategoryId: string
    - CategoryName: name
  + **Category-Movie** (tự động gen trong DB, ko có trong Models) (mỗi Movie cx có nhiều Category)
    - CategoryId
    - MovieId
  + **Actor**
    - ConnectionId: int (thể hiện kết nối giữa 2 table)
    - ActorId: int
    - ActorName: string
  + **Actor-Movie** (tự động gen trong DB, ko có trong Models)
    - ActorId
    - MovieId

## {Kiên} [API] Admin

* CRUD User
  + Get list of users
  + Get user by id
  + Search user by name
  + Add new user
  + Update user
  + Delete user
* CRUD Movie
  + List Movie: Paging by 10
  + Add movie
  + Update movie
  + Delete movie

## {Thông} [API] Movies (Client)

* Client:
  + List Category
  + List Movie: Paging by 5
  + Movie Detail

## {An} [API] [WebClient] Authentication

* Login
* Sign up
* Lưu dữ liệu vào Database: Password -> Hashing
  + Sử dụng thuật toán **SHA256**

## {Huyền} [WebClient] Watch Movie

* User – Xem phim (Page):
  + Pause/Continue
  + Tua (tiến/lùi 10s)
  + Chỉnh âm lượng
  + Thay đổi kích cỡ:
    - Mặc định
    - Full Screen
    - Picture-in-picture
  + Quay lại Home

## {Huyền} Design Front-end:

* Logo:
  + Navbar
  + Favicon
* Style Front-end (Page):
  + Login
  + Sign up
  + Watch Movie

## {Kiên} [API] (for Client)

* Role:
  + Get List Role (For User Management: Add/Update)
  + Get Role by ID
* Actor:
  + Get all Actors
  + Get Actors by MovieID (For Movie Detail)
* Category:
  + Get List Categories by MovieID (For Movie List – Client)

## {Kiên} [WebClient] User Management (Admin)

* CRUD User:
  + List: Paging
  + Add/Update:
    - Hủy: bấm nút Back về List
    - Hiển thị RoleName thay vì RoleID
    - Thành công / Thất bại: Hiển thị thông báo (Bootstrap Modal)
  + Add:
    - Mã hóa Password
  + Update User: ko update Password
  + Delete: Confirm Modal
  + Search:
    - Nếu không ra kết quả -> Hiển thị “No Data”
    - Clear Search: bấm vào thì xóa kết quả search (reload page)

## {Giang} Database Seeding

* DbContext: Seed cho tất cả các Table

## {Giang} [WebClient] (User) Home Page

* ***Featured Movie***: hiển thị 1 phim nổi bật
  + Poster
  + Description
  + Watch
* ***New releases***: Danh sách các phim mới   
  (thực ra là tất cả các phim, làm để Demo Paging vì DB có ít phim quá)
  + Hover:
    - Name
    - Watch
* ***Marvels***: Danh sách các phim Siêu anh hùng của Marvel

## {An} [API]-[WebClient] (User) Authentication

* ***Login***
* ***Sign up***
* ***Login with Facebook***

## {Kiên} [WebClient] (User) Update Profile

* Nhận vào UserID (từ URL) => Call API => Lấy dữ liệu về User => Gán lên các <input> để ng dùng sửa
  + Ko đc phép sửa Role
  + Ko sửa Password ở đây
* => Submit -> Show Modal:
  + Thành công -> OK -> về index
  + Thất bại -> OK -> Tắt Modal
* Button Back -> về index (hủy update)