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