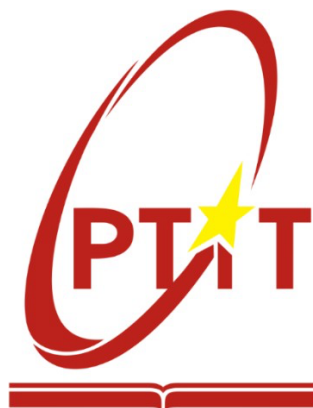


**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----oOo-----



**BÀI TẬP LỚN  
LẬP TRÌNH VỚI PYTHON**

**NHÓM 12**

**Đề tài: Thiết kế trang Web Playlist Video**

**Giảng viên: Nguyễn Trọng Khánh**

**Nhóm môn học: Nhóm 01**

**Thành viên nhóm:**

**Nguyễn Mạnh Cường      B22DCCN099**

**Đặng Ngọc Bách      B22DCCN051**

**Lê Thành Đạt      B22DCCN189**

**Trần Xuân Bắc      B22DCCN075**

**HÀ NỘI – 2024**

# MỤC LỤC

LỜI NÓI ĐẦU .....	2
I. Giới thiệu: .....	3
1. Lý do chọn đề tài .....	3
2. Mô tả hệ thống:.....	3
3. Đối tượng người dùng:.....	3
II. Phân công nhiệm vụ .....	4
III. Thiết kế hệ thống:.....	5
1. Thư viện sử dụng: .....	5
2. Các file và thư mục của dự án:.....	5
3. backend.py .....	6
a. Class Video, Playlist, User và Admin:.....	6
b. Class PlaylistManager:.....	6
c. Class UserManager: .....	7
d. Class AdminManager .....	8
4. app.py:.....	8
a. Chức năng cho người dùng: .....	8
b. Chức năng cho quản trị viên: .....	9
c. Các chức năng phụ trợ.....	9
5. template:.....	9
IV. Demo hệ thống .....	11
1. Đăng nhập/Đăng kí.....	11
2. User: .....	12
3. Admin:.....	16

## LỜI NÓI ĐẦU

Python là một trong những ngôn ngữ lập trình phổ biến và mạnh mẽ, được sử dụng rộng rãi trong các lĩnh vực như phát triển phần mềm, xử lý dữ liệu, trí tuệ nhân tạo và nhiều ứng dụng khác. Với cú pháp rõ ràng, dễ đọc và thư viện phong phú, Python mang lại sự tiện lợi và hiệu quả cao trong quá trình phát triển ứng dụng.

Dự án "Playlist Video" được thực hiện nhằm áp dụng các kiến thức đã học về ngôn ngữ Python để xây dựng một ứng dụng giúp người dùng lưu và quản lý các video yêu thích. Qua dự án này, chúng em không chỉ hiểu rõ hơn về cách tổ chức, xử lý và tương tác dữ liệu trong môi trường lập trình mà còn rèn luyện kỹ năng phân tích, thiết kế hệ thống, và giải quyết các vấn đề thực tiễn.

Sản phẩm đã được hoàn thiện với sự nỗ lực và cẩn thận, tuy nhiên, không tránh khỏi những thiếu sót và hạn chế. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ cô để có thể hoàn thiện dự án một cách tốt nhất.

Chúng em xin gửi lời cảm ơn sâu sắc tới cô Nguyễn Quỳnh Chi đã tận tình giảng dạy và hướng dẫn trong suốt quá trình học tập và thực hiện dự án. Những kiến thức mà cô truyền đạt là nền tảng vững chắc để chúng em hoàn thành bài tập lớn này cũng như áp dụng trong công việc sau này.

Chúng em xin chân thành cảm ơn!

## **I. Giới thiệu:**

### **1. Lý do chọn đề tài:**

Trong cuộc sống hiện đại, video đã trở thành một phần không thể thiếu, đóng vai trò quan trọng không chỉ trong giải trí mà còn hỗ trợ đắc lực cho học tập, làm việc và phát triển cá nhân.

Sự phát triển mạnh mẽ của các nền tảng chia sẻ video như YouTube đã mang đến lượng nội dung khổng lồ, đa dạng về chủ đề và thể loại. Tuy nhiên, việc quản lý và lưu trữ các video yêu thích lại trở thành thách thức lớn đối với người dùng, đặc biệt khi họ muốn hệ thống hóa hoặc truy cập nhanh chóng vào nội dung đã chọn lọc.

Nhiều người dùng thường cảm thấy bất tiện khi phải phụ thuộc vào tính năng danh sách phát (playlist) của từng nền tảng. Những khó khăn này không chỉ làm giảm hiệu quả trải nghiệm người dùng mà còn cản trở việc khai thác tối ưu các giá trị mà video mang lại.

Do đó, một hệ thống quản lý danh sách phát video độc lập là giải pháp cần thiết, giúp người dùng dễ dàng lưu trữ, sắp xếp, và truy cập vào các video yêu thích mọi lúc, mọi nơi. Hệ thống này sẽ không chỉ giúp quản lý video một cách khoa học mà còn cung cấp các chức năng bổ trợ như đánh giá, mô tả, và chia sẻ danh sách phát với cộng đồng, từ đó nâng cao trải nghiệm sử dụng trong các mục đích cá nhân và chuyên nghiệp.

### **2. Mô tả hệ thống:**

Dự án *Playlist Video* cung cấp một hệ thống đơn giản nhưng hiệu quả, giúp người dùng lưu trữ, tổ chức, và xem lại các video yêu thích của mình. Không chỉ vậy, ứng dụng còn mang lại khả năng chia sẻ playlist với bạn bè hoặc cộng đồng, tạo nên một môi trường chia sẻ nội dung dễ dàng và tiện lợi.

Ứng dụng được thiết kế với hệ thống thân thiện, dễ sử dụng, kết hợp giữa tính năng cơ bản và mở rộng nhằm đáp ứng nhiều nhu cầu của người dùng. Hệ thống cũng tích hợp các vai trò quản lý như Admin để điều phối nội dung và hỗ trợ người dùng.

### **3. Đối tượng người dùng:**

Ứng dụng hướng tới hai đối tượng người dùng với các chức năng cơ bản sau:

#### **Người dùng (User):**

- Đăng ký và đăng nhập tài khoản.
- Tạo danh sách phát riêng tư hoặc công khai.
- Thêm và xóa video khỏi danh sách phát.
- Xem và lưu danh sách phát yêu thích.
- Chỉnh sửa playlist.

**Quản trị viên (Admin):**

- Quản lý tài khoản người dùng.
- Xóa hoặc kiểm duyệt các nội dung không phù hợp.
- Xem và quản lý danh sách phát được chia sẻ công khai.
- Cập nhật thông tin hệ thống và duy trì hoạt động ổn định.

## II. Thiết kế hệ thống:

### 1. Thư viện sử dụng:

#### Flask (trong app.py):

- Flask là một web framework nhẹ, dùng để xây dựng ứng dụng web.
- Các thành phần được sử dụng:
  - Flask: Tạo ứng dụng Flask.
  - render\_template: Kết nối các template HTML với backend để hiển thị giao diện.
  - request: Xử lý HTTP requests (GET, POST).
  - redirect: Chuyển hướng người dùng đến các route khác.
  - url\_for: Xây dựng URL động.
  - session: Quản lý trạng thái đăng nhập (lưu thông tin người dùng trong session).
  - flash: Hiển thị thông báo cho người dùng (thành công, lỗi).

#### OS (trong app.py và backend.py):

- Mô tả: Thư viện tiêu chuẩn của Python để làm việc với hệ điều hành.
- Các thành phần được sử dụng:
  - os.environ['PYTHONIOENCODING']: Đặt mã hóa đầu ra để trợ tiếng Việt.
  - os.path.exists: Kiểm tra sự tồn tại của file trước khi đọc/ghi.

#### Webbrowser (trong backend.py):

- Thư viện để mở trình duyệt web từ chương trình Python.
- Các thành phần được sử dụng:
  - webbrowser.open: Dùng để mở link video trong trình duyệt khi người dùng phát video.

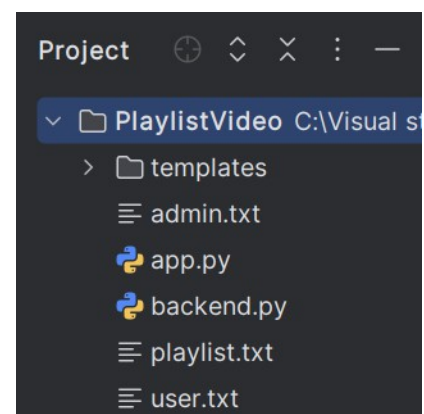
### 2. Các file và thư mục của dự án:

#### templates:

- Là nơi lưu trữ các tệp HTML. Đây là nơi chứa các template (giao diện) được sử dụng để hiển thị nội dung trang web cho người dùng.

#### app.py:

- Dùng thư viện Flask xử lý giao diện người dùng và các vấn đề liên quan đến trang web.
- Bao gồm các route điều hướng và logic xử lý HTTP request/response, như đăng nhập, đăng ký, tạo, xem, cập nhật playlist, và quản lý người dùng.



### **backend.py:**

- Chứa các lớp và logic cốt lõi xử lý dữ liệu như video, playlist, người dùng, và quản trị viên.
- Quản lý đọc/ghi dữ liệu từ tệp (playlist.txt, user.txt, admin.txt) và thực hiện các thao tác như thêm, xóa, cập nhật playlist hoặc người dùng.

### **admin.txt, user.txt, playlist.txt:**

- Dữ liệu được lưu trữ trong các tệp .txt, bao gồm user.txt, admin.txt, và playlist.txt.
- Các tệp này đóng vai trò lưu trữ dữ liệu ứng dụng, như thông tin người dùng, quản trị viên, và playlist, để truy xuất và xử lý trong các chức năng của chương trình.

## **3. backend.py:**

### **a. Class Video, Playlist, User và Admin:**

#### **Class Video:**

- Gồm có thuộc tính chủ đề và link video.
- Phương thức:
  - `__init__(self, chude, link)`: Khởi tạo video với chủ đề và link.
  - `open(self)`: Mở video trong trình duyệt mặc định của người dùng.

#### **Class Playlist:**

- Thuộc tính: ID, tacgia, mota, danhgia, list\_videos và status (public hoặc private).

#### **Class User:**

- Thuộc tính: ID, username, password, email, sdt, check.

#### **Class Admin:**

- Thuộc tính: ID, username, password, email, sdt.

### **b. Class PlaylistManager:**

#### **Đọc dữ liệu playlist:**

- **get\_public\_playlists(user\_id)**: Lấy danh sách các playlist công khai của tất cả người dùng, ngoại trừ người dùng có user\_id.
- **get\_user\_playlist(user\_id)**: Lấy playlist của một người dùng cụ thể dựa trên user\_id.
- **get\_user\_playlists(user\_id)**: Lấy tất cả các playlist của người dùng cụ thể.
- **get\_all\_playlists()**: Lấy toàn bộ playlist trong hệ thống từ file playlist.txt.
- **read\_playlist\_from\_text()**: Đọc tất cả các playlist từ file playlist.txt và trả về danh sách các đối tượng Playlist.

- **read\_videos\_from\_text(data):** Chuyển đổi dữ liệu từ file thành danh sách các đối tượng Video.

#### Ghi và cập nhật dữ liệu playlist:

- **write\_videos\_to\_text(list\_videos, file):** Ghi danh sách video của một playlist vào file.
- **write\_playlist\_to\_text(playlist, user\_id, check):** Ghi hoặc cập nhật một playlist: Nếu check = True: Cập nhật playlist hiện có của người dùng. Nếu check = False: Thêm playlist mới.

#### Kiểm tra và xóa playlist:

- **check\_user\_playlist(user\_id):** Kiểm tra xem người dùng đã có playlist hay chưa.
- **delete\_playlist\_from\_text(playlist):** Xóa một playlist khỏi file playlist.txt.
- **Quản lý video trong playlist:**
- **play\_video(playlist, video\_index):** Phát một video cụ thể trong playlist, mở video trong trình duyệt.
- **add\_video(playlist):** Thêm một video mới vào playlist.
- **update\_playlist(playlist):** Cập nhật thông tin của playlist, như tác giả, mô tả, hoặc đánh giá.
- **delete\_video(playlist):** Xóa một video khỏi playlist theo chỉ số.

#### Hàm phụ trợ:

- **read\_video\_from\_keyboard():** Nhập thông tin video mới (chủ đề và link) từ người dùng qua giao diện dòng lệnh.
- **print\_videos(list\_videos):** In danh sách video của một playlist ra màn hình.
- **select\_in\_range(prompt, min, max):** Nhận đầu vào từ người dùng và đảm bảo giá trị nằm trong khoảng [min, max].

#### c. Class UserManager:

- **sign\_up\_user(username, password, email, sdt):** Đăng ký người dùng mới, kiểm tra tên đăng nhập có tồn tại không.
- **save\_user(user):** Lưu thông tin người dùng vào file user.txt.
- **read\_list\_user\_from\_text():** Đọc danh sách người dùng từ file user.txt.
- **sign\_in(username, password):** Xác thực đăng nhập của người dùng.
- **get\_user\_id(username):** Lấy ID người dùng dựa trên tên đăng nhập.
- **get\_user\_by\_username(username):** Lấy đối tượng User dựa trên tên đăng nhập.

```
class UserManager:
    @staticmethod
    def sign_up_user(username, password, email, sdt): ...

    @staticmethod
    def save_user(user): ...

    @staticmethod
    def read_list_user_from_text(): ...

    @staticmethod
    def sign_in(username, password): ...

    @staticmethod
    def get_user_id(username): ...

    @staticmethod
    def get_user_by_username(username): ...

    @staticmethod
    def ban_user(user): ...

    @staticmethod
    def unban_user(user): ...
```



- **ban\_user(user):** Cấm một người dùng bằng cách đổi trạng thái VALID thành INVALID trong file.
- **unban\_user(user):** Bỏ cấm một người dùng bằng cách đổi trạng thái INVALID thành VALID.

**d. Class AdminManager:**

- **sign\_up\_admin(username, password, email, sdt):** Đăng ký quản trị viên mới.
- **save\_admin(admin):** Lưu thông tin quản trị viên vào file admin.txt.
- **read\_list\_admin\_from\_text():** Đọc danh sách quản trị viên từ file admin.txt.
- **sign\_in(username, password):** Xác thực đăng nhập của quản trị viên.

```
class AdminManager:
    @staticmethod
    def sign_up_admin(username, password, email, sdt): ...

    @staticmethod
    def save_admin(admin): ...

    @staticmethod
    def read_list_admin_from_text(): ...

    @staticmethod
    def sign_in(username, password): ...
```

**4. app.py:**

Các hàm trong app.py tập trung vào xử lý HTTP request và logic hiển thị giao diện cho người dùng, kết hợp với backend để quản lý dữ liệu và thực hiện các thao tác như đăng nhập, đăng ký, tạo/sửa/xóa playlist, và quản lý người dùng và dùng để chạy chương trình.

**a. Chức năng cho người dùng:**

- **Đăng nhập (/)**
  - Người dùng hoặc quản trị viên có thể đăng nhập vào hệ thống.
  - Kiểm tra thông tin đăng nhập và phân quyền (user/admin).
- **Đăng ký (/sign\_up)**
  - Người dùng có thể đăng ký tài khoản mới với thông tin như username, password, email, và số điện thoại.
- **Đăng xuất (/logout)**
  - Xóa thông tin phiên đăng nhập và quay về trang chính.
- **Hiển thị menu người dùng (/user\_menu)**
  - Truy cập các chức năng liên quan đến playlist của người dùng.
- **Quản lý playlist cá nhân:**
  - Tạo playlist (/create\_playlist): Tạo mới hoặc ghi đè playlist.
  - Xem playlist (/show\_playlist): Hiển thị danh sách video trong playlist, phát video.
  - Thêm video (/add\_video): Thêm một video mới vào playlist.
  - Xóa video (/delete\_video): Xóa video khỏi playlist.
  - Cập nhật playlist (/update\_playlist): Chỉnh sửa thông tin playlist (tác giả, mô tả, đánh giá, trạng thái).
- **Xem playlist công khai (/public\_playlists)**

- Hiển thị các playlist công khai từ người dùng khác.
- b. Chức năng cho quản trị viên:**
  - **Hiển thị menu admin (/admin\_menu)**
    - Truy cập các chức năng quản trị như quản lý người dùng và playlist.
  - **Quản lý playlist người dùng:**
    - Hiển thị playlist (/admin\_show\_playlist): Xem tất cả playlist của người dùng.
    - Xóa playlist (/admin\_delete\_playlist): Xóa một playlist bất kỳ.
  - **Quản lý người dùng (/admin\_show\_users)**
    - Hiển thị danh sách người dùng đã đăng ký.
  - **Cấm/Bỏ cấm người dùng (/admin\_ban\_unban\_user)**
    - Quản lý trạng thái tài khoản người dùng (VALID hoặc INVALID).
- c. Các chức năng phụ trợ:**
  - **Chuyển đổi URL video (convert\_to\_embed\_url)**
    - Biến đổi link YouTube sang dạng nhúng (embed) để phát trực tiếp trên trình duyệt.

## 5. template:

- **home.html:** tạo giao diện chính của trang chủ, nơi người dùng bắt đầu trải nghiệm ứng dụng. Trang này chào đón người dùng và cung cấp các tùy chọn như đăng nhập hoặc chuyển sang trang đăng ký tài khoản.
- **sign\_up.html:** tạo giao diện để người dùng đăng ký tài khoản mới. Nó cung cấp form cho người dùng nhập thông tin như tên đăng nhập, mật khẩu, email và số điện thoại, đảm bảo quá trình đăng ký diễn ra thuận tiện và nhanh chóng.
- **add\_playlist.html:** được thiết kế để tạo giao diện thêm playlist mới vào tài khoản của người dùng. Giao diện hiển thị các trường nhập thông tin như tên playlist và mô tả, giúp người dùng dễ dàng quản lý nội dung cá nhân.
- **add\_video.html:** tạo giao diện để người dùng thêm video vào playlist. Người dùng có thể nhập chủ đề và link liên kết video để thêm video vào playlist của mình.

```

v templates
  <> add_playlist.html
  <> add_video.html
  <> admin_menu.html
  <> admin_show_playlist.html
  <> ban_unban_user.html
  <> create_playlist.html
  <> delete_playlist.html
  <> delete_video.html
  <> edit_playlist.html
  <> home.html
  <> play_playlist.html
  <> playlist_list.html
  <> public_playlists.html
  <> show_playlist.html
  <> show_users.html
  <> sign_up.html
  <> user_menu.html

```

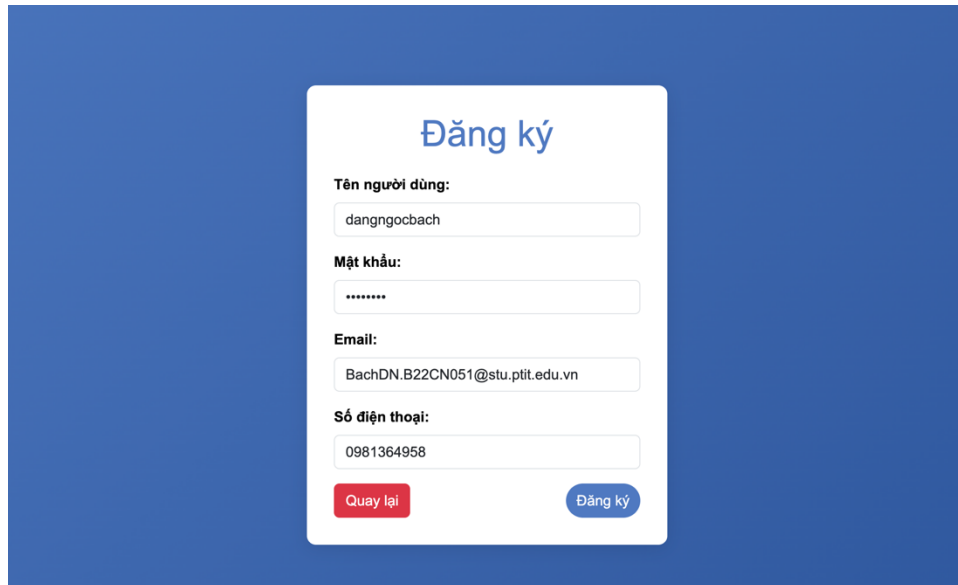
- **create\_playlist.html:** tạo giao diện để người dùng tạo một playlist mới từ đầu. Người dùng có thể nhập thông tin như tác giả , mô tả , đánh giá , trạng thái (Public hoặc Private), và thêm các video ban đầu để hoàn thiện playlist.
- **delete\_video.html:** tạo giao diện để người dùng xóa video khỏi một playlist cụ thể. Người dùng có thể chọn video để xóa một cách dễ dàng.
- **edit\_playlist.html:** tạo giao diện để người dùng chỉnh sửa thông tin của playlist. Người dùng có thể cập nhật các trường như tác giả , mô tả, đánh giá và trạng thái của playlist
- **play\_playlist.html:** tạo giao diện để người dùng phát các video trong một playlist. Trang hiển thị danh sách video kèm giao diện trình phát, cho phép người dùng chọn video và xem nội dung trực tiếp trên trang.
- **public\_playlist.html:** tạo giao diện để người dùng chọn danh sách các playlist công khai do cộng đồng chia sẻ. Người dùng có thể xem nội dung các playlist và phát video từ những playlist được chia sẻ.
- **user\_menu.html:** tạo giao diện menu chính của người dùng sau khi đăng nhập. Nó cung cấp các liên kết đến các chức năng như tạo playlist , hiển thị danh sách playlist , thêm video vào playlist , cập nhật playlist , xóa video khỏi playlist, xem playlist công khai và đăng xuất khỏi tài khoản.
- **admin\_menu.html:** tạo giao diện cho Menu Quản Trị Viên, nơi admin có thể quản lý nội dung ứng dụng như hiển thị danh sách playlist và người dùng, xóa playlist, quản lý trạng thái tài khoản (ban/unban), đồng thời hỗ trợ đăng xuất.
- **admin\_show\_playlist.html:** tạo giao diện hiển thị danh sách các playlist hiện có trong hệ thống, kèm theo thông tin chi tiết như ID, tác giả, mô tả, đánh giá, trạng thái và danh sách video trong mỗi playlist.
- **ban\_undban\_user.html:** được thiết kế để quản trị viên quản lý trạng thái ban/unban của người dùng trong hệ thống.
- **delete\_playlist.html:** là một giao diện quản lý được thiết kế để hỗ trợ xóa các playlist trong cơ sở dữ liệu.
- **playlist\_list.html:** cung cấp giao diện thân thiện để quản trị viên xem thông tin về các playlist.
- **show\_playlist.html:** là một giao diện mở rộng, hỗ trợ phát playlist trực tiếp trên trang web.
- **show\_users.html:** là một giao diện trực quan dành cho quản trị viên để xem thông tin các người dùng.

### III. Demo hệ thống:

#### 1. Đăng nhập/Đăng kí:

##### Đăng kí:

Nếu chưa có tài khoản, người dùng cần ấn vào nút đăng kí để đăng kí một tài khoản mới. Khi tạo tài khoản mới người dùng cần nhập tên tài khoản, mật khẩu, email và số điện thoại rồi nhấn Đăng kí sau khi đã hoàn thành.

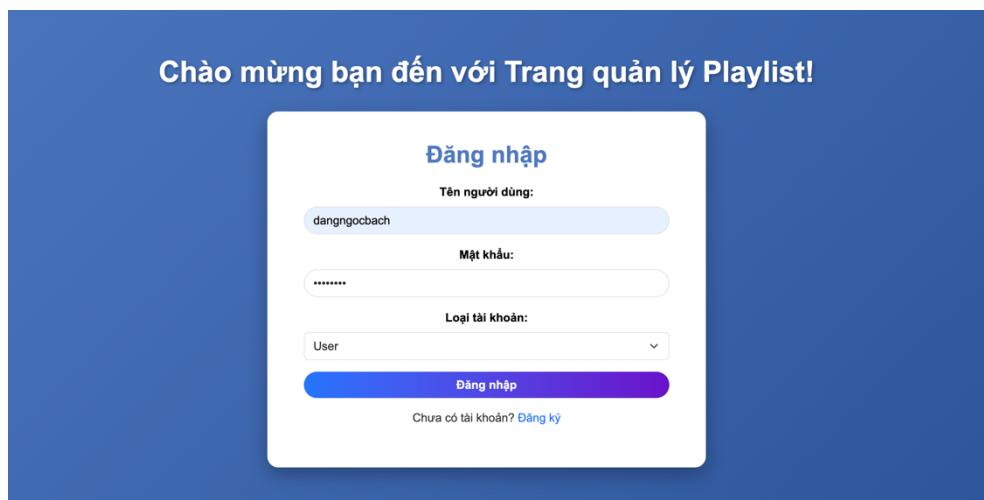


The registration form is titled "Đăng ký" and is set against a blue background. It contains the following fields and buttons:

- Tên người dùng: dangngocbach
- Mật khẩu: (masked with asterisks)
- Email: BachDN.B22CN051@stu.ptit.edu.vn
- Số điện thoại: 0981364958
- Buttons: Quay lại (red), Đăng ký (blue)

##### Đăng nhập:

Nếu đã có tài khoản, người dùng sẽ nhập tên tài khoản, mật khẩu và lựa chọn loại tài khoản phù hợp và nhấn Đăng nhập.



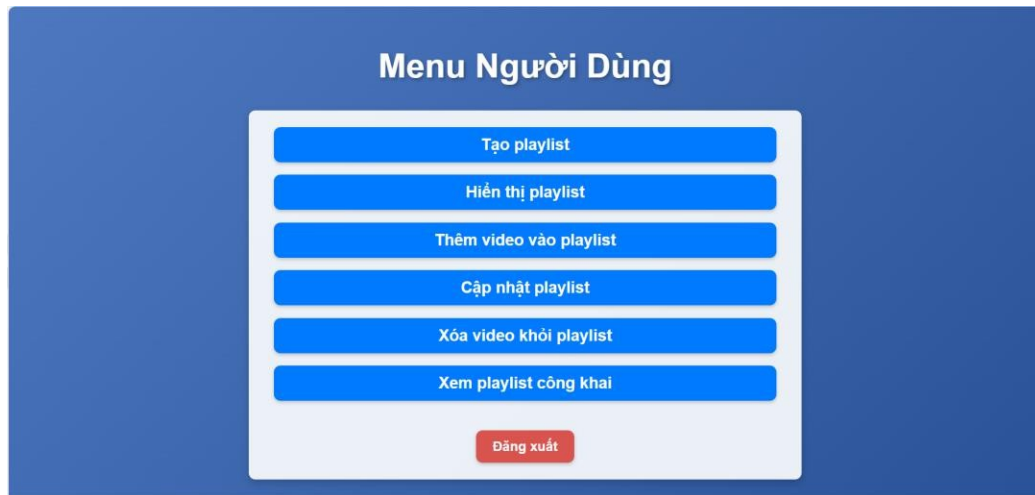
The login form is titled "Đăng nhập" and is set against a blue background. It contains the following fields and buttons:

- Tên người dùng: dangngocbach
- Mật khẩu: (masked with asterisks)
- Loại tài khoản: User (dropdown menu)
- Buttons: Đăng nhập (blue/purple gradient)
- Link: Chưa có tài khoản? [Đăng ký](#)

## 2. User:

### Menu người dùng:

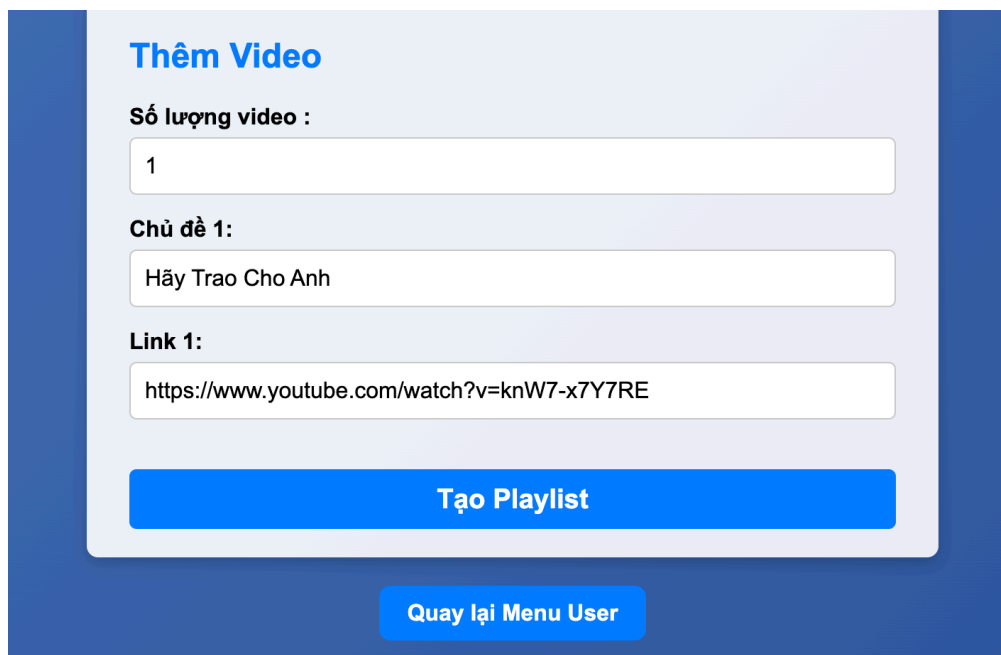
Sau khi đăng nhập, nếu tài khoản người dùng là tài khoản User, trang web sẽ hiện lên giao diện với các chức năng của User để người dùng lựa chọn.



### Tạo playlist:

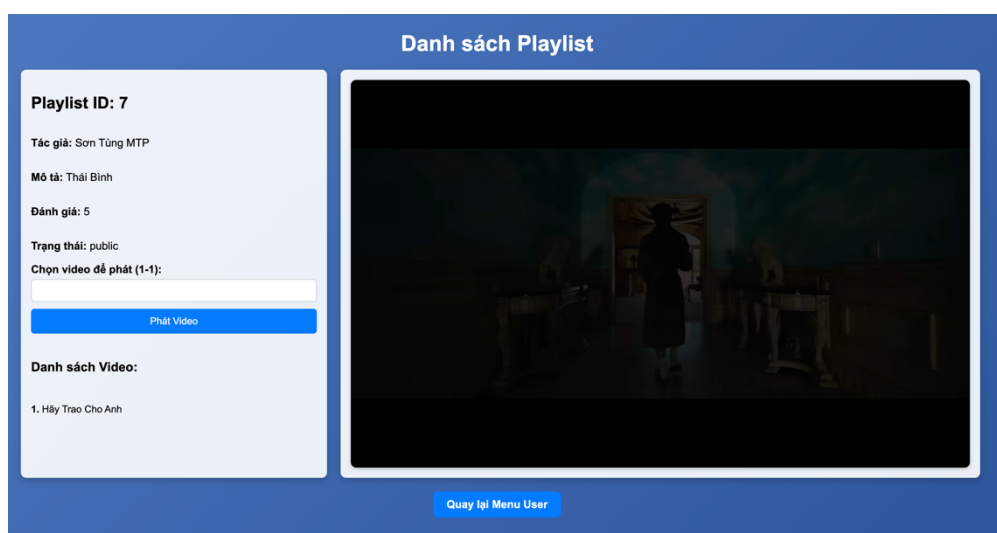
Nếu chưa có Playlist, người dùng cần ấn vào Tạo playlist để tạo Playlist mới của mình. Trong trang Tạo playlist, người dùng cần nhập: tên tác giả, mô tả, đánh giá (là một số nguyên từ 1 – 5), trạng thái (Public nếu muốn người khác có thể xem được playlist của mình hoặc Private nếu không muốn người khác xem được playlist của mình).

Sau đó nhập vào số video muốn thêm vào playlist. Mỗi video người dùng cần nhập vào chủ đề và link của video đó. Sau khi hoàn thành các bước trên, người dùng ấn Tạo playlist.



### Hiển thị Playlist:

Trong trang hiển thị Playlist, người dùng có thể xem các video trong playlist của mình bằng việc điền một số vào trong ô Chọn video để phát. Trang sẽ phát video trong danh sách video bên dưới tương ứng với số được điền vào ô.



### Thêm video vào Playlist:

Trong trang này, bạn cần nhập vào chủ đề và link của video mà bạn muốn thêm sau đó ấn Thêm video.

## Thêm Video vào Playlist

**Chủ đề:**

**Link:**

Thêm Video

Quay lại Menu User

### Cập nhật Playlist:

Trong trang này, người dùng có thể sửa tác giả, mô tả, đánh giá và trạng thái Playlist của mình.

## Sửa Playlist

**Tác giả:**

**Mô tả:**

**Đánh giá:**

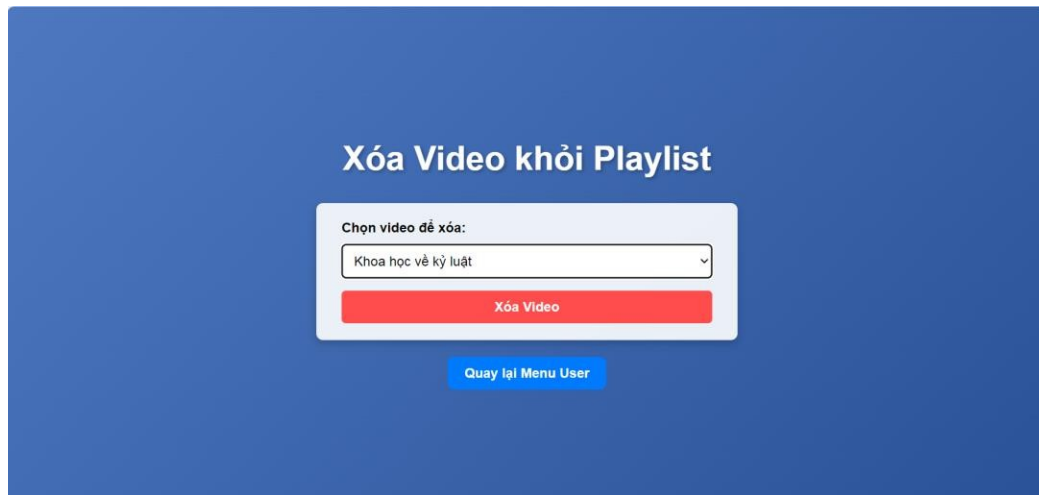
**Trạng thái:**

Lưu

Quay lại Menu User

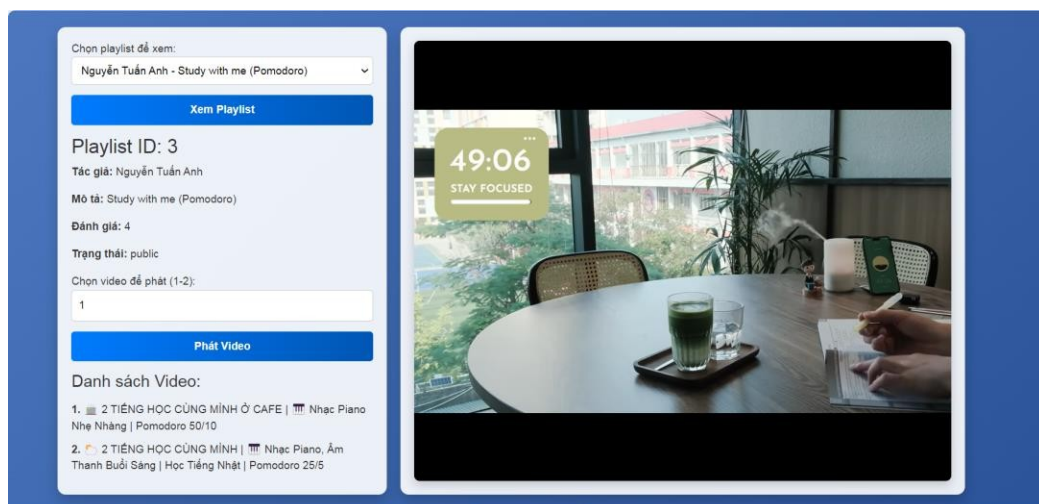
### Xóa video khỏi Playlist:

Trong trang này người dùng sẽ chọn chủ đề của video mà mình muốn xóa trong danh sách sau đó ấn Xóa video.



### Xem Playlist công khai:

Trong trang này, người dùng có thể xem được những Playlist của người khác nếu họ để chế độ Public. Sau khi chọn Playlist để một người dùng, Playlist của họ sẽ hiện ra bên dưới với đầy đủ thông tin của Playlist. Người dùng có thể ấn chọn video để xem.

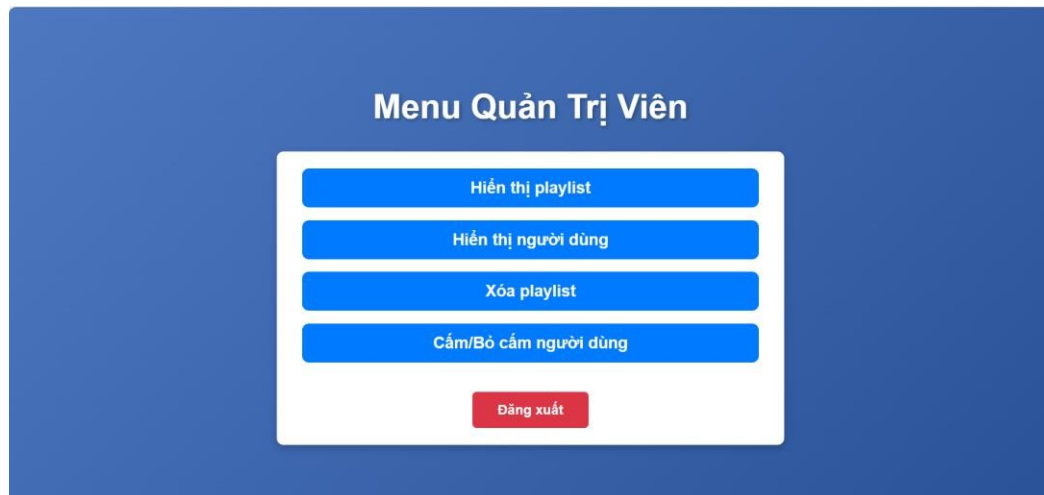




### 3. Admin:

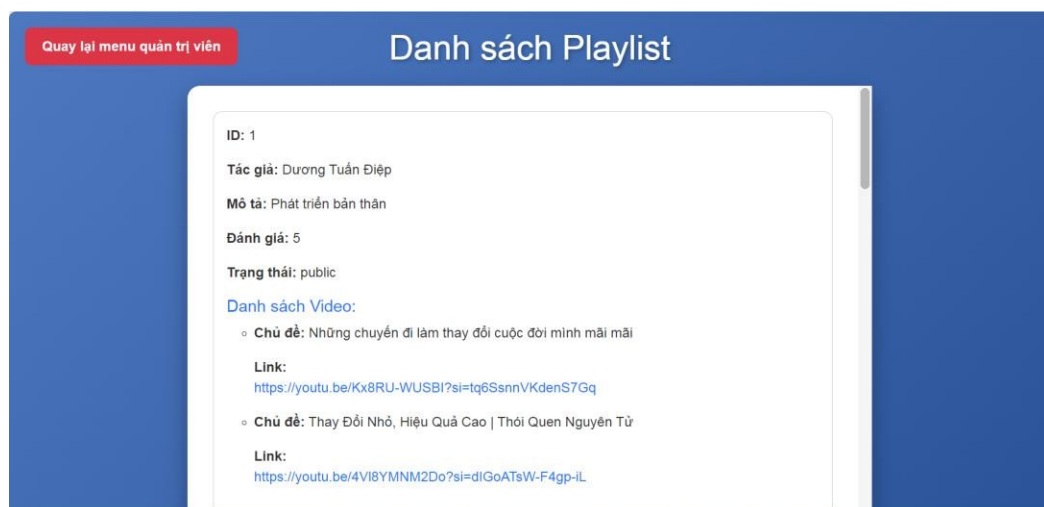
#### Menu Quản trị viên:

Nếu người dùng là quản trị viên, sau khi đăng nhập, giao diện quản trị viên sẽ hiện ra với đầy đủ chức năng của quản trị viên.



#### Hiển thị Playlist:

Trong trang này, quản trị viên có thể xem được toàn bộ Playlist của tất cả các người dùng.



### Hiển thị người dùng:

Trong trang này, quản trị viên có thể xem được các thông tin của người dùng bao gồm: ID, Tên, Email, Số điện thoại và Trạng thái.

[Quay lại menu quản trị viên](#)

### Danh sách Người Dùng

ID	Tên người dùng	Email	Số điện thoại	Trạng thái
1	Dương Tuấn Điệp	duongtuandiep2@gmail.com	0867545881	VALID
2	Nguyễn Tuấn Anh	nguyentuananh@gmail.com	0972357312	VALID
3	Đinh Hoàng Hải	hoanghaidinh1124@gmail.com	0987678533	VALID
4	Đào Đức Hiếu	daoduchieu@gmail.com	0398690016	VALID
5	Nguyễn Xuân Đạt	datnger123@gmail.com	0397524973	VALID

### Xóa Playlist:

Trong trang này, quản trị viên có thể xóa Playlist của người dùng nếu Playlist đó chứa những video có nội dung không phù hợp.

[Quay lại menu quản trị viên](#)

### Xóa Playlist

**ID: 1**

Xóa

Tác giả: Dương Tuấn Điệp

Mô tả: Phát triển bản thân

Đánh giá: 5

**ID: 3**

Xóa

Tác giả: Nguyễn Tuấn Anh

Mô tả: Study with me (Pomodoro)

Đánh giá: 4

**ID: 4**

Xóa

### Cấm/Bỏ cấm người dùng:

Trong trang này, người dùng có thể cấm hoặc bỏ cấm đối với các tài khoản người dùng.

Chức năng này khác xóa Playlist ở chỗ khi xóa Playlist người dùng vẫn có thể tạo một Playlist mới còn khi cấm tài khoản thì người dùng không thể đăng nhập bằng tài khoản bị cấm.

[Quay lại menu quản trị viên](#)

### Cấm/Bỏ cấm người dùng

ID	Tên người dùng	Email	Số điện thoại	Trạng thái	Hành động
1	Dương Tuấn Điệp	duongtuandiep2@gmail.com	0867545881	VALID	<a href="#">Cấm</a>
2	Nguyễn Tuấn Anh	nguyentuananh@gmail.com	0972357312	VALID	<a href="#">Cấm</a>
3	Đinh Hoàng Hải	hoanghaidinh1124@gmail.com	0987678533	VALID	<a href="#">Cấm</a>
4	Đào Đức Hiếu	daoduchieu@gmail.com	0398690016	VALID	<a href="#">Cấm</a>
5	Nguyễn Xuân Đạt	datnger123@gmail.com	0397524973	VALID	<a href="#">Cấm</a>