**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA ĐA PHƯƠNG TIỆN** Shape, square

Description automatically generated



**BÁO CÁO NGHIÊN CỨU KHOA HỌC**

**ĐỀ TÀI:**

**Nghiên cứu thử nghiệm lập trình Python nhúng thủy vân số để bảo vệ bản quyền sản phẩm đa phương tiện**

**Mã số: 02-SV-2024-ĐPT**

**Giảng viên hướng dẫn**: TS. Trần Quý Nam

**Sinh viên thực hiện**:

|  |  |
| --- | --- |
| Đỗ Trung Hiếu | B21DCPT011 |
| Đỗ Tuấn Minh | B21DCPT023 |
| Đỗ Đăng Tuân | B21DCPT225 |
|  |  |
| **Hà Nội, 2024** | |

**LỜI CẢM ƠN**

Đầu tiên, em xin gửi lời cảm ơn chân thành và sự kính trọng tới các thầy cô giáo trong Khoa Đa phương tiện, đặc biệt là **TS Trần Quý Nam**, người đã tận tâm hướng dẫn chúng em trong suốt quá trình thực hiện và hoàn thiện đề tài nghiên cứu khoa học của mình..

Em và toàn thể nhóm rất vinh dự khi được thầy Nam làm giáo viên hướng dẫn cho đề tài nghiên cứu khoa học của mình. Thầy là một người thầy tận tâm, nhiệt huyết và giàu kinh nghiệm trong lĩnh vực nghiên cứu khoa học. Thầy đã luôn tận tình giúp đỡ em trong suốt quá trình nghiên cứu, từ những bước đầu tiên tìm hiểu đề tài, đến việc xây dựng kế hoạch nghiên cứu, thu thập dữ liệu, phân tích kết quả và hoàn thiện báo cáo. Thầy đã dành rất nhiều thời gian để giải đáp thắc mắc của em, giúp em tìm ra những hướng đi đúng đắn trong nghiên cứu.

Nhờ sự hướng dẫn tận tình của thầy Nam, chúng em đã có thể hoàn thành đề tài nghiên cứu của mình một cách tốt nhất. Em cũng như các bạn trong nhóm đã học hỏi được rất nhiều từ thầy về kiến thức chuyên môn, kỹ năng nghiên cứu khoa học. Thay mặt cho cả nhóm, Em xin chân thành cảm ơn thầy vì tất cả những gì thầy đã dành cho chúng em cũng như các sinh viên mà thầy đã và đang dìu dắt.

Tuy đây không phải là lần đầu tiên nhóm chúng em thực hiện nghiên cứu đề tài, tìm hiểu về lĩnh vực sáng tạo trong nghiên cứu khoa học song kiến thức của nhóm còn hạn chế và thiếu sót. Vì vậy, nhóm em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để đề tài nghiên cứu của em được hoàn thiện hơn.

Cuối cùng, em xin kính chúc quý thầy cô trong Khoa Đa phương tiện thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

**Mục lục**

[❖ CHƯƠNG 1: NGHIÊN CỨU, KHẢO SÁT 4](#_Toc179571335)

[1.1 Đặt vấn đề: 4](#_Toc179571336)

[1.2 Lý do lựa chọn đề tài 5](#_Toc179571337)

[1.3 Kết chương 6](#_Toc179571338)

[❖ CHƯƠNG 2: PHÂN TÍCH VÀ PHÁT TRIỂN ỨNG DỤNG THỦY VÂN SỐ 8](#_Toc179571339)

[2.1. Khái niệm và vai trò của thủy vân số 8](#_Toc179571342)

[2.2. Các phương pháp tích hợp nhúng thủy vân số 11](#_Toc179571343)

[2.3. Phát triển ứng dụng nhúng thủy vân số bằng Python 13](#_Toc179571344)

[2.4. Công cụ và thư viện hỗ trợ lập trình 14](#_Toc179571345)

[2.5. Kết chương 17](#_Toc179571346)

[❖ CHƯƠNG 3 : XÂY DỰNG ỨNG DỤNG VÀ THỬ NGHIỆM 18](#_Toc179571347)

[3.1. Mô tả yêu cầu hệ thống 18](#_Toc179571349)

[3.2. Mô tả nghiệp vụ của các chức năng 19](#_Toc179571350)

[3.3. Thông tin các đối tượng cần xử lý 21](#_Toc179571351)

[3.4. Kiến trúc hệ thống 22](#_Toc179571352)

[3.5. Quy trình hoạt động của hệ thống 28](#_Toc179571353)

[3.6. Cấu trúc và quy trình nhúng thủy vân 29](#_Toc179571354)

[3.7. Thử nghiệm hệ thống trên các sản phẩm đa phương tiện 31](#_Toc179571355)

[3.8. Hình ảnh ứng dụng 32](#_Toc179571356)

[❖ CHƯƠNG 4: KẾT LUẬN 36](#_Toc179571357)

[4.1 Yêu cầu đặt ra: 36](#_Toc179571358)

[4.2 Kết quả đã thực hiện nghiên cứu: 36](#_Toc179571359)

[4.3 Hướng phát triển tiếp: 37](#_Toc179571360)

## ❖ CHƯƠNG 1: NGHIÊN CỨU, KHẢO SÁT

### Đặt vấn đề:

#### Khảo sát hiện trạng lập trình nhúng thủy vân số để bảo vệ bản quyền sản phẩm đa phương tiện bằng ngôn ngữ lập trình Python

* Trong bối cảnh sự tăng trưởng nhanh chóng của nội dung số, việc bảo vệ bản quyền các sản phẩm đa phương tiện (hình ảnh, video, tài liệu văn bản) đã trở thành một nhu cầu thiết yếu. Nhúng thủy vân số được xem là một trong những phương pháp hiệu quả nhất để bảo vệ bản quyền. Python, với khả năng linh hoạt và được hỗ trợ bởi nhiều thư viện mạnh mẽ, đã trở thành một ngôn ngữ được ưa chuộng để phát triển các ứng dụng nhúng thủy vân số.
* Trong phần này, chúng em sẽ khảo sát các phương pháp lập trình nhúng thủy vân số hiện có bằng Python, đánh giá mức độ hiệu quả, độ phổ biến và khả năng áp dụng trong thực tế. Python cung cấp các thư viện mạnh mẽ như **OpenCV, Pillow, MoviePy**, cho phép thực hiện nhúng thủy vân vào các loại dữ liệu đa phương tiện một cách dễ dàng và hiệu quả. Việc sử dụng Python để lập trình nhúng thủy vân số giúp giảm thiểu chi phí phát triển và tăng tính khả dụng của ứng dụng trong các ngành công nghiệp khác nhau

#### Phân tích các lợi ích khi lập trình nhúng thủy vân số bằng Python:

* **Khả năng xử lý linh hoạt**: Python hỗ trợ nhiều thư viện như OpenCV, Pillow và MoviePy, cho phép nhúng thủy vân số vào các đối tượng đa phương tiện một cách nhanh chóng và hiệu quả. Các thư viện này cung cấp các hàm xử lý hình ảnh, video mạnh mẽ, hỗ trợ nhiều định dạng dữ liệu và dễ dàng tùy chỉnh theo nhu cầu của người dùng.
* **Dễ dàng tiếp cận**: Python là ngôn ngữ có cú pháp dễ hiểu, giúp các nhà phát triển dễ tiếp cận và phát triển ứng dụng. Điều này đặc biệt hữu ích cho những người mới bắt đầu hoặc không có kinh nghiệm lập trình nhiều, giúp họ nhanh chóng nắm bắt và áp dụng vào các dự án thực tế.
* **Tính tích hợp cao**: Python có thể tích hợp với nhiều công nghệ khác nhau như Flask để xây dựng web, Docker để triển khai đa nền tảng, và Kubernetes để quản lý container. Điều này giúp Python trở thành một lựa chọn lý tưởng cho việc phát triển các ứng dụng có khả năng mở rộng và dễ dàng triển khai trên các môi trường khác nhau.
* **Khả năng mở rộng**: Python có khả năng kết nối và tích hợp với các công cụ AI và học máy, giúp nâng cao tính năng bảo mật và khả năng tự động hóa trong việc phát hiện và bảo vệ bản quyền.

#### Phân tích những khó khăn khi lập trình nhúng thủy vân số với ngôn ngữ lập trình Python:

* Dù Python mang lại nhiều lợi ích, nhưng việc lập trình nhúng thủy vân số bằng ngôn ngữ này cũng đối mặt với nhiều khó khăn:
  + **Hiệu suất xử lý**: Python có hiệu suất thấp hơn so với các ngôn ngữ khác như C/C++. Việc nhúng thủy vân số vào video độ phân giải cao có thể táo ra độ trễ.
  + **Khó khăn trong việc xử lý đa luồng**: Python bị hạn chế bởi GIL (Global Interpreter Lock), điều này làm giới hạn tính năng xử lý song song đối với các nhiệm vụ xử lý nặng như video. Điều này có thể ảnh hưởng đến khả năng mở rộng và hiệu suất của ứng dụng, đặc biệt khi phải xử lý nhiều yêu cầu đồng thời.
  + **Bảo mật**: Python có tính linh hoạt cao nhưng có nguy cơ bị lợi bỏ đỉnh khi xử lý dữ liệu từ nhiều nguồn khác nhau. Việc nhúng thủy vân số phải được kiểm soát để tránh bị thao tác bất hợp pháp.
  + **Thiếu các thư viện tối ưu cho xử lý thời gian thực**: Dù Python có nhiều thư viện mạnh mẽ, nhưng việc xử lý thời gian thực cho video hoặc hình ảnh có thể gặp khó khăn do hiệu suất không cao. Điều này đòi hỏi phải có sự kết hợp với các ngôn ngữ khác hoặc sử dụng các thư viện được tối ưu hóa tốt hơn.

### Lý do lựa chọn đề tài

* Thủy vân số là một phương pháp bảo vệ bản quyền kỹ thuật số bằng cách nhúng thông tin nhận dạng (chẳng hạn như logo, chữ ký số hoặc các thông tin khác) trực tiếp vào nội dung đa phương tiện như hình ảnh, video, và tài liệu văn bản. Điều này giúp bảo vệ quyền sở hữu trí tuệ và cho phép truy xuất nguồn gốc của tài liệu khi có sự vi phạm bản quyền xảy ra. Khái niệm thủy vân số xuất phát từ việc sử dụng các dấu nhận dạng bí mật được nhúng vào dữ liệu mà không làm ảnh hưởng đến trải nghiệm người dùng đối với nội dung đó.
* Một trong những lợi ích chính của việc sử dụng nhúng thủy vân số là khả năng ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Đối với ngành công nghiệp truyền thông, nhúng thủy vân số có thể bảo vệ nội dung số khỏi các hành vi sao chép bất hợp pháp. Trong giáo dục và nghiên cứu, kỹ thuật này giúp bảo vệ các tài liệu số quan trọng, tránh bị chỉnh sửa hoặc sao chép mà không có sự cho phép. Hơn nữa, trong lĩnh vực y tế, nhúng thủy vân số cũng có thể được sử dụng để bảo vệ các hình ảnh y khoa, đảm bảo rằng dữ liệu bệnh nhân được bảo mật và duy trì tính toàn vẹn.
* Ứng dụng nhúng thủy vân số được tích hợp, triển khai trên nền tảng web, cho phép người dùng dễ dàng thêm nhúng thủy vân số vào hình ảnh, văn bản và video một cách nhanh chóng và hiệu quả. Để tối ưu hiệu quả của giải pháp này, chúng em đã quyết định sử dụng Python để phát triển một ứng dụng tích hợp nhúng thủy vân số, tận dụng khả năng xử lý linh hoạt và sự hỗ trợ từ các thư viện mạnh mẽ của Python như OpenCV, MoviePy và Pillow.

### Kết chương

* Chương này đã trình bày bối cảnh và tầm quan trọng của việc tích hợp nhúng thủy vân số vào các sản phẩm kỹ thuật số, đồng thời xác định nhu cầu cấp thiết đối với giải pháp tự động và hiệu quả cho vấn đề này. Việc sử dụng Python để phát triển ứng dụng nhúng thủy vân số là lựa chọn phù hợp nhờ khả năng xử lý mạnh mẽ và sự phong phú của các thư viện hỗ trợ. Đây sẽ là nền tảng cho các bước nghiên cứu và phát triển tiếp theo được trình bày trong các chương sau. Việc sử dụng các thư viện mã nguồn mở như OpenCV, MoviePy và Pillow không chỉ cho phép tăng cường khả năng tùy chỉnh và mở rộng của ứng dụng mà còn giúp giảm chi phí phát triển. Tầm quan trọng của việc bảo vệ bản quyền và xác thực nguồn gốc tài sản kỹ thuật số thông qua giải pháp nhúng thủy vân số là động lực quan trọng để tiếp tục phát triển và tối ưu hóa hệ thống này.
* Bên cạnh đó, nghiên cứu này cũng đặt ra nhiều câu hỏi quan trọng cần giải quyết, chẳng hạn như làm thế nào để tối ưu hóa quá trình nhúng thủy vân số mà không làm suy giảm chất lượng của dữ liệu gốc? Những phương pháp nào có thể giúp tăng cường bảo mật và khả năng phát hiện nhúng thủy vân số khi dữ liệu bị xâm phạm hoặc chỉnh sửa? Đây là những câu hỏi quan trọng sẽ được giải quyết trong các chương tiếp theo.

## ❖ CHƯƠNG 2: PHÂN TÍCH VÀ PHÁT TRIỂN ỨNG DỤNG THỦY VÂN SỐ



### Khái niệm và vai trò của thủy vân số

#### Khái niệm

* Nhúng thủy vân số là một kỹ thuật bảo vệ bản quyền bằng cách nhúng thông tin nhận dạng (như logo, văn bản) vào trong các sản phẩm kỹ thuật số như hình ảnh, văn bản và video. Nhúng thủy vân số không chỉ giúp nhận dạng nguồn gốc của sản phẩm mà còn ngăn chặn hành vi sao chép và phân phối trái phép. Việc phát triển một ứng dụng nhúng thủy vân số linh hoạt, có khả năng tích hợp với nhiều loại sản phẩm kỹ thuật số là yêu cầu cần thiết trong bối cảnh hiện tại.

#### Vai trò

* Bảo vệ bản quyền: Bảo vệ các tác phẩm đa phương tiện, để ngăn chặn sao chép trái phép
* Xác thực nguồn gốc: Cho phép xác nhận nguồn gốc và tác giả của tài liệu, đảm bảo tính minh bạch và xác thực của nội dung. Điều này đặc biệt quan trọng đối với các tài liệu nhạy cảm hoặc có giá trị thương mại cao.
* Theo dõi vi phạm: Truy xuất nguồn gốc tài liệu khi có vi phạm bản quyền xảy ra. Thông qua thông tin nhúng trong nội dung, người sở hữu có thể theo dõi và xác định các vi phạm một cách dễ dàng.

#### Đặc điểm

* **Tính bền bỉ (Robustness)**: Nhúng thủy vân số cần phải tồn tại ngay cả khi dữ liệu đã bị xử lý qua nhiều kỹ thuật khác nhau, chẳng hạn như nén, cắt xén, hoặc chuyển đổi định dạng.
* **Tính an toàn (Security)**: Thủy vân phải được nhúng một cách bí mật, không thể dễ dàng bị phát hiện hoặc bị gỡ bỏ mà không sử dụng các công cụ đặc biệt.
* **Tính khó nhận biết (Imperceptibility)**: Không làm thay đổi cảm nhận của người dùng đối với nội dung. Thủy vân số phải được nhúng một cách tinh tế để người dùng không nhận ra sự khác biệt so với nội dung gốc.

#### Phân loại

* **Nhúng thủy vân số rõ ràng**: Đây là loại thủy vân có thể dễ dàng nhìn thấy bằng mắt thường. Ví dụ điển hình là các logo hoặc thông điệp bản quyền được chèn trực tiếp vào hình ảnh, video, hoặc tài liệu văn bản. Nhúng thủy vân số rõ ràng có ưu điểm là làm cho người xem ngay lập tức nhận biết được quyền sở hữu của nội dung, do đó giúp ngăn chặn hành vi sao chép trái phép. Đây là một phương pháp hiệu quả để công khai quyền sở hữu và thường được sử dụng bởi các công ty truyền thông, báo chí, và các trang web chia sẻ nội dung. Tuy nhiên, nhúng thủy vân số rõ ràng có nhược điểm là dễ bị loại bỏ hoặc chỉnh sửa bởi các công cụ chỉnh sửa ảnh/video phổ biến, chẳng hạn như Photoshop, làm giảm khả năng bảo vệ bản quyền nếu kẻ xâm phạm có kỹ năng và ý định xóa bỏ dấu hiệu bản quyền.
* **Nhúng thủy vân số ẩn**: Ngược lại với nhúng thủy vân số rõ ràng, nhúng thủy vân số ẩn không thể nhìn thấy bằng mắt thường và yêu cầu các công cụ hoặc thuật toán đặc biệt để phát hiện. Thủy vân số ẩn thường được nhúng vào các phần tử không dễ nhận biết của dữ liệu, chẳng hạn như các giá trị pixel hoặc các hệ số biến đổi tần số. Ưu điểm của loại này là khó bị loại bỏ hoặc phát hiện bởi những người không có kiến thức chuyên môn và công cụ đặc biệt, giúp bảo vệ bản quyền một cách bí mật và hiệu quả hơn. Nhúng thủy vân số ẩn có tính bảo mật cao, phù hợp cho các tình huống cần theo dõi và xác minh nguồn gốc của dữ liệu sau khi đã chia sẻ hoặc phân phối rộng rãi. Tuy nhiên, nhược điểm của nhúng thủy vân số ẩn là có thể bị suy giảm nếu nội dung trải qua các quá trình biến đổi như nén hoặc mã hóa, do đó việc lựa chọn thuật toán nhúng và vị trí nhúng là rất quan trọng để duy trì tính toàn vẹn của thủy vân.

#### So sánh với các phương pháp bảo vệ bản quyền khác

* Nhúng thủy vân số không chỉ là một trong những phương pháp bảo vệ bản quyền số, mà còn có thể so sánh với nhiều phương pháp bảo vệ bản quyền kỹ thuật số khác như mã hóa, DRM (Quản lý quyền kỹ thuật số), và dấu vết kỹ thuật số (digital fingerprinting).
* So với mã hóa, nhúng thủy vân số cung cấp một phương pháp bảo vệ bản quyền mà không làm hạn chế quyền truy cập của người dùng cuối. Trong khi mã hóa bảo vệ nội dung bằng cách ẩn đi thông tin cho đến khi được giải mã, nhúng thủy vân số nhúng thông tin nhận dạng trực tiếp vào nội dung mà không làm thay đổi cách thức truy cập. Điều này có nghĩa là người dùng vẫn có thể xem nội dung, nhưng dấu hiệu bảo vệ bản quyền vẫn được giữ nguyên. Tuy nhiên, mã hóa có lợi thế về khả năng bảo vệ mạnh mẽ hơn trước các hành vi sao chép trái phép, đặc biệt khi dữ liệu cần được bảo mật tuyệt đối trong quá trình truyền tải.
* DRM là một công cụ bảo vệ bản quyền khác thường được sử dụng để hạn chế các hành vi sử dụng trái phép thông qua các hạn chế về quyền truy cập. DRM thường được triển khai ở cấp hệ thống hoặc phần mềm, giúp quản lý và kiểm soát ai có quyền truy cập vào nội dung số. Tuy nhiên, DRM có một số hạn chế, bao gồm gây khó chịu cho người dùng hợp pháp và dễ bị các phần mềm crack vượt qua. Trong khi đó, nhúng thủy vân số không hạn chế quyền truy cập mà thay vào đó tập trung vào việc đánh dấu bản quyền và theo dõi nguồn gốc nội dung, cho phép xác định trách nhiệm khi xảy ra vi phạm.
* Dấu vết kỹ thuật số (digital fingerprinting) là một phương pháp khác liên quan đến việc nhúng thông tin nhận dạng vào nội dung số, nhưng nó thường được sử dụng để theo dõi và xác định nguồn gốc của nội dung khi có nhiều bản sao được phân phối cho các người dùng khác nhau. Dấu vết kỹ thuật số có thể được sử dụng để xác định người dùng cụ thể đã rò rỉ hoặc chia sẻ nội dung trái phép. Nhúng thủy vân số có tính tổng quát hơn, nhắm đến việc bảo vệ bản quyền và xác định quyền sở hữu, trong khi dấu vết kỹ thuật số nhắm đến việc kiểm soát và theo dõi hành vi người dùng.
* Nhìn chung, nhúng thủy vân số có lợi thế là không gây cản trở đến trải nghiệm người dùng hợp pháp và có thể duy trì dấu hiệu bảo vệ ngay cả khi nội dung trải qua các quá trình biến đổi. Tuy nhiên, nhúng thủy vân số có thể bị loại bỏ hoặc thay đổi bởi những người có kỹ năng và công cụ phù hợp, và do đó cần kết hợp với các biện pháp bảo vệ khác để đảm bảo hiệu quả bảo mật cao nhất. Sự kết hợp của các phương pháp như mã hóa, DRM, và nhúng thủy vân số có thể cung cấp một hệ thống bảo vệ bản quyền toàn diện hơn, đáp ứng tốt các yêu cầu đa dạng trong việc bảo vệ tài sản kỹ thuật số.

### Các phương pháp tích hợp nhúng thủy vân số

* Các bước tích hợp nhúng thủy vân số trong Python đòi hỏi sự kết hợp giữa các kỹ thuật xử lý ảnh/video và các phương pháp bảo mật để đảm bảo hiệu quả bảo vệ và tính toàn vẹn của nội dung số. Việc sử dụng các thư viện như OpenCV, Pillow, MoviePy và ReportLab giúp quá trình này trở nên linh hoạt và dễ dàng triển khai trên nhiều loại dữ liệu khác nhau.Có nhiều phương pháp để tích hợp nhúng thủy vân số vào hình ảnh, văn bản và video. Đối với hình ảnh và video, các phương pháp phổ biến bao gồm nhúng trực tiếp nhúng thủy vân số lên nội dung bằng cách sử dụng các thư viện Python như OpenCV hoặc MoviePy. Phương pháp này đảm bảo nhúng thủy vân số có thể dễ dàng nhìn thấy hoặc có thể ẩn đi mà không làm ảnh hưởng đáng kể đến chất lượng sản phẩm. Việc sử dụng các thuật toán nhúng thủy vân số trong miền tần số cũng là một lựa chọn hiệu quả để bảo vệ bản quyền mà không làm giảm chất lượng của tệp gốc. Có nhiều phương pháp để tích hợp nhúng thủy vân số vào hình ảnh, văn bản và video. Trong phần này, chúng em sẽ phân tích kỹ hơn các bước thực hiện trong Python, bao gồm logic xử lý và cách sử dụng các thư viện quan trọng để thực hiện nhúng thủy vân số
* Phương pháp nhúng thủy vân số cho hình ảnh và video
  + Sử dụng thư viện OpenCV và Pillow:
    - **Tiền xử lý dữ liệu**: Bước đầu tiên là tiền xử lý hình ảnh hoặc video. Điều này bao gồm việc tải tệp tin, thay đổi kích thước, hoặc chuyển đổi định dạng màu. Trong Python, chúng em sử dụng thư viện OpenCV để đọc và xử lý hình ảnh.
    - **Nhúng thủy vân số rõ ràng**: Nhúng thủy vân số rõ ràng có thể được thực hiện bằng cách chèn một hình ảnh hoặc văn bản lên hình ảnh hoặc khung hình video (Đối với nhúng thủy vân số rõ ràng, việc xác định vị trí và độ trong suốt của thủy vân rất quan trọng để không ảnh hưởng đến trải nghiệm của người dùng.)
    - **Nhúng thủy vân số ẩn**: Đối với nhúng thủy vân số ẩn, các thông tin bảo mật được nhúng vào các kênh màu hoặc các thành phần tần số. Chúng em có thể sử dụng biến đổi DCT (Discrete Cosine Transform) hoặc DWT (Discrete Wavelet Transform) để nhúng thủy vân vào các thành phần tần số thấp hoặc cao (Biến đổi DCT và DWT giúp thủy vân khó bị phát hiện và loại bỏ, đảm bảo tính bảo mật cao.)
  + Sử dụng thư viện MoviePy:
    - MoviePy được sử dụng để xử lý và thêm nhúng thủy vân số vào video. Chúng em có thể thêm nhúng thủy vân số rõ ràng vào video bằng cách chèn một lớp hình ảnh hoặc văn bản vào mỗi khung hình
    - MoviePy cho phép xử lý từng khung hình và thêm nhúng thủy vân số theo cách tùy chỉnh, đảm bảo rằng nhúng thủy vân số có thể dễ dàng tích hợp vào nội dung video mà không ảnh hưởng đến chất lượng.
* Phương pháp nhúng thủy vân số cho văn bản
  + Sử dụng thư viện ReportLab:
    - Đối với các tài liệu văn bản, nhúng thủy vân số thường được thực hiện bằng cách thêm các ký tự ẩn hoặc nhúng các đoạn văn bản ngắn vào tài liệu. ReportLab là một thư viện Python mạnh mẽ cho phép tạo và chỉnh sửa các tệp PDF, giúp tích hợp nhúng thủy vân số vào tài liệu văn bản
    - Nhúng thủy vân số vào tài liệu văn bản cần đảm bảo không làm thay đổi nội dung chính và không ảnh hưởng đến khả năng đọc của tài liệu.
    - **Nhúng thủy vân số ẩn cho văn bản**: Một phương pháp khác là thêm các ký tự ẩn hoặc thay đổi các dấu cách trong văn bản để mã hóa thông tin bảo vệ. Điều này giúp bảo vệ bản quyền một cách bí mật mà người dùng cuối không nhận ra sự khác biệt.

### Phát triển ứng dụng nhúng thủy vân số bằng Python

#### Giới thiệu tổng quan

* Ứng dụng được phát triển với mục tiêu chính là thử nghiệm và phát triển việc nhúng thủy vân số lên các tệp đa phương tiện như hình ảnh, video và tài liệu PDF. Ứng dụng cho phép người dùng tải lên các tệp cần bảo vệ, tùy chỉnh nội dung thủy vân, và nhúng trực tiếp vào sản phẩm mà không ảnh hưởng đến chất lượng của tệp gốc.
* Người dùng có thể lựa chọn nhúng thủy vân dưới dạng văn bản hoặc hình ảnh, tùy chỉnh vị trí, kích thước, và độ trong suốt của thủy vân để phù hợp với yêu cầu sử dụng. Sau khi quá trình nhúng thủy vân hoàn tất, tệp đã được nhúng thủy vân số sẽ được trả về cho người dùng.
* Ứng dụng này được phát triển với mục tiêu không chỉ nhằm mục đích thử nghiệm nhúng thủy vân số vào các sản phẩm đa phương tiện mà còn cung cấp thêm các tính năng quản lý tệp, giúp người dùng dễ dàng thao tác và bảo vệ bản quyền nội dung của mình.

#### Xác định các chức năng chính của hệ thống

* Ứng dụng nhúng thủy vân số được phát triển với một loạt các chức năng chính nhằm hỗ trợ người dùng trong việc nhúng thủy vân vào các sản phẩm đa phương tiện, đồng thời cung cấp các công cụ quản lý tệp một cách tiện lợi. Các chức năng chính bao gồm:
  + Tải lên tệp đa phương tiện:
    - Cho phép người dùng tải lên các tệp hình ảnh, video và tài liệu PDF từ thiết bị cá nhân. Hệ thống sẽ hỗ trợ các định dạng tệp phổ biến như JPEG, PNG, MP4, PDF,… giúp mở rộng khả năng xử lý cho nhiều loại nội dung khác nhau.
  + Nhúng thủy vân vào tệp:
    - Người dùng có thể lựa chọn nhúng thủy vân dưới dạng văn bản hoặc hình ảnh. Tính năng này cho phép tùy chỉnh vị trí, kích thước, màu sắc, và độ trong suốt của thủy vân trước khi nhúng vào tệp.
  + Xem trước thủy vân:
    - Cung cấp chức năng xem trước nội dung nhúng thủy vân trên tệp trước khi người dùng xác nhận và lưu thay đổi. Điều này giúp đảm bảo rằng thủy vân được nhúng đúng vị trí và theo ý muốn của người dùng.
  + Quản lý tệp đã tải lên:
    - Người dùng có thể xem lại các tệp đã tải lên và thực hiện các thao tác như sửa đổi hoặc xóa tệp. Mỗi tệp được lưu trữ với thông tin về thời gian tải lên và trạng thái đã nhúng thủy vân hay chưa.
  + Xác thực người dùng:
    - Hệ thống tích hợp tính năng xác thực người dùng để đảm bảo rằng chỉ có những người dùng đã đăng nhập và có quyền mới có thể truy cập và thao tác trên tệp đa phương tiện.

### Công cụ và thư viện hỗ trợ lập trình

#### Thư viện chính cho nhúng thủy vân số

* Pillow:
  + Pillow là thư viện xử lý hình ảnh, được sử dụng để nhúng thủy vân vào tệp hình ảnh (JPEG, PNG). Thư viện cung cấp các công cụ để thêm văn bản hoặc hình ảnh lên tệp, tùy chỉnh vị trí, kích thước và màu sắc của thủy vân mà không làm giảm chất lượng của ảnh gốc.
* MoviePy:
  + MoviePy là thư viện dùng để chỉnh sửa video, cho phép nhúng thủy vân vào các tệp video với định dạng phổ biến như MP4, MOV và AVI. MoviePy hỗ trợ việc thêm thủy vân dưới dạng văn bản hoặc hình ảnh vào video mà vẫn giữ nguyên chất lượng của video gốc.
* ReportLab và PyPDF2:
  + ReportLab giúp tạo và chỉnh sửa tài liệu PDF, bao gồm nhúng thủy vân vào các trang PDF.
  + PyPDF2 hỗ trợ đọc và chỉnh sửa các tệp PDF hiện có, cho phép nhúng thủy vân vào các tài liệu đã tồn tại mà không cần tạo mới từ đầu.
* Tempfile:
  + Tempfile là thư viện dùng để tạo và quản lý các tệp tạm thời khi xử lý dữ liệu trước khi lưu trữ chính thức, đảm bảo quá trình nhúng thủy vân không làm ảnh hưởng đến tệp gốc.

#### Backend (BE)

* Django Rest Framework (DRF):
  + DRF là nền tảng để xây dựng các API RESTful, cung cấp các tính năng như tải lên tệp, nhúng thủy vân và trả về tệp đã xử lý. DRF hỗ trợ xác thực người dùng và quản lý các yêu cầu xử lý từ phía người dùng một cách linh hoạt.
* MongoDB:
  + MongoDB là cơ sở dữ liệu NoSQL được sử dụng để lưu trữ dữ liệu người dùng và các tệp đa phương tiện. Cơ sở dữ liệu này giúp lưu trữ linh hoạt và hiệu quả khi xử lý khối lượng lớn dữ liệu không có cấu trúc, đặc biệt là các tệp đa phương tiện.
* Google OAuth:
  + Google OAuth được tích hợp để xác thực người dùng, đảm bảo rằng chỉ người dùng đã xác thực mới có quyền truy cập và thực hiện thao tác trên hệ thống.
* Django File System Storage:
  + Django File System Storage quản lý việc lưu trữ các tệp đã tải lên, đảm bảo tệp hình ảnh, video và PDF được lưu trữ và có thể truy cập để xử lý nhúng thủy vân khi cần.
* VPS (Virtual Private Server):
  + VPS được sử dụng để host và triển khai toàn bộ backend của ứng dụng, bao gồm Django Rest Framework và MongoDB. Việc sử dụng VPS giúp đảm bảo hiệu suất và bảo mật cao, với khả năng quản lý server một cách độc lập và linh hoạt. Ngoài ra, VPS cung cấp môi trường ổn định và đáng tin cậy cho việc xử lý khối lượng lớn các yêu cầu từ người dùng trong thời gian thực.

#### Frontend (FE)

* Vite:
  + Vite được sử dụng làm công cụ build và phát triển frontend. Vite cung cấp một môi trường phát triển nhanh và hiệu quả, giúp cải thiện tốc độ biên dịch và tối ưu hóa mã nguồn trong quá trình phát triển.
* Vue 3:
  + Vue 3 là framework chính được sử dụng cho frontend, cho phép tạo các giao diện người dùng tương tác và thân thiện. Vue 3 giúp quản lý các trạng thái động, tương tác và xử lý dữ liệu từ backend thông qua API.
* Axios:
  + Axios là thư viện dùng để gửi các yêu cầu HTTP từ frontend đến backend. Trong ứng dụng này, Axios được sử dụng để giao tiếp với API RESTful của DRF, giúp người dùng có thể tải lên tệp, nhúng thủy vân, và nhận lại kết quả xử lý từ server một cách dễ dàng và hiệu quả.
* Element Plus:
  + Element Plus là một thư viện giao diện người dùng (UI library) được xây dựng cho Vue 3, cung cấp một tập hợp các thành phần (components) sẵn sàng sử dụng để giúp các nhà phát triển tạo ra các ứng dụng web với giao diện người dùng chuyên nghiệp và nhất quán. Nó là phiên bản nâng cấp và tối ưu hóa của Element UI (thư viện UI phổ biến cho Vue 2), được thiết kế lại để tận dụng các tính năng mới của Vue 3 và hỗ trợ tốt hơn cho các dự án hiện đại.

### Kết chương

* Qua các phân tích về thủy vân số cũng như phương pháp nhúng thủy vân số, kết hợp với nghiên cứu và ứng dụng các công cụ, thư viện Python phù hợp, để phát triển hệ ứng dụng web nhúng thủy vân cho các tệp đa phương tiện.

## ❖ CHƯƠNG 3 : XÂY DỰNG ỨNG DỤNG VÀ THỬ NGHIỆM



### Mô tả yêu cầu hệ thống

* Việc thiết kế và triển khai hệ thống nhúng thủy vân số yêu cầu phải đáp ứng các chức năng quan trọng nhằm đảm bảo khả năng bảo vệ bản quyền thông qua việc nhúng thủy vân vào tệp đa phương tiện, đồng thời cung cấp trải nghiệm người dùng trực quan và dễ sử dụng. Hệ thống phải hỗ trợ tải lên tệp, nhúng thủy vân dưới dạng văn bản hoặc hình ảnh, cho phép xem trước, và cung cấp công cụ quản lý tệp một cách hiệu quả. Các yêu cầu chi tiết của hệ thống bao gồm:
  + **Tải lên tệp đa phương tiện**: Người dùng phải có khả năng tải lên các tệp đa phương tiện, bao gồm các loại tệp như hình ảnh (JPEG, PNG), video (MP4, MOV). Hệ thống cần hỗ trợ nhiều định dạng tệp để mở rộng phạm vi ứng dụng trong nhiều ngành khác nhau, từ truyền thông đến giáo dục. Quá trình tải lên cần được tối ưu hóa để đảm bảo tốc độ và sự ổn định, đặc biệt khi xử lý các tệp lớn.
  + **Nhúng thủy vân**: Hệ thống phải cho phép người dùng nhúng thủy vân dưới dạng văn bản hoặc hình ảnh lên các tệp đã tải lên. Thủy vân văn bản có thể là một chuỗi ký tự như tên tác giả, thông tin bản quyền hoặc logo. Thủy vân hình ảnh thường là các biểu tượng hoặc đồ họa nhỏ được đặt lên nội dung. Cả hai loại thủy vân này cần có khả năng tùy chỉnh về vị trí, kích thước, màu sắc và độ trong suốt. Điều này đảm bảo rằng thủy vân có thể được nhúng một cách linh hoạt vào tệp mà không làm ảnh hưởng đến trải nghiệm người dùng khi xem nội dung.
  + **Chức năng xem trước**: Sau khi chọn thủy vân, hệ thống cần cung cấp tính năng xem trước, cho phép người dùng kiểm tra cách thức hiển thị của thủy vân trên tệp trước khi lưu thay đổi. Chức năng này đảm bảo rằng thủy vân được đặt đúng vị trí, với kích thước và độ trong suốt phù hợp, giúp người dùng tự tin hơn trong việc thực hiện các thao tác nhúng thủy vân vào tệp.
  + **Quản lý tệp**: Hệ thống cần cung cấp khả năng quản lý tệp cho người dùng, bao gồm xem lại các tệp đã tải lên, chỉnh sửa, hoặc xóa các tệp không cần thiết. Người dùng có thể xem các thông tin về tệp như thời gian tải lên, trạng thái nhúng thủy vân, và các đặc điểm của tệp (kích thước, định dạng). Chức năng quản lý này giúp người dùng theo dõi được quá trình bảo vệ bản quyền của mình một cách hiệu quả.
  + **Xác thực người dùng**: Để đảm bảo tính bảo mật và phân quyền truy cập, hệ thống cần tích hợp xác thực người dùng, chẳng hạn như thông qua Google OAuth. Chỉ những người dùng đã được xác thực và đăng nhập mới có thể truy cập vào các chức năng của hệ thống như tải lên tệp và nhúng thủy vân. Điều này giúp ngăn chặn truy cập trái phép và bảo vệ dữ liệu của người dùng.

### Mô tả nghiệp vụ của các chức năng

* Đăng nhập
  + **Mục tiêu:** Cho phép người dùng truy cập vào ứng dụng.
  + Hoạt động:
    - Người dùng sử dụng tài khoản google để đăng nhập.
    - Google xác thực người dùng.
    - Nếu hợp lệ, Google sẽ gửi lại thông tin người dùng cho hệ thống để cập nhật vào cơ sở dữ liệu.
    - Nếu không hợp lệ, hệ thống báo lỗi và yêu cầu người dùng đăng nhập lại.
* Đăng xuất
  + **Mục tiêu:** Kết thúc phiên làm việc của người dùng.
  + Hoạt động:
    - Người dùng chọn chức năng đăng xuất.
    - Hệ thống xóa thông tin đăng nhập của người dùng.
    - Người dùng được chuyển hướng về trang đăng nhập.
* Tải lên tệp đa phương tiện
  + **Mục tiêu:** Tệp đa phương tiện của người dùng được tải lên.
  + Hoạt động:
    - Người dùng chọn chức năng tải lên tệp đa phương tiện.
    - Người dùng chọn tệp đa phương tiện muốn tải lên.
    - Hệ thống tiến hành lưu tệp của người dùng sau khi người dùng xác nhận.
    - Hệ thống thông báo cho người dùng.
* Xem chi tiết tệp đã tải lên
  + **Mục tiêu:** Người dùng xem được chi tiết các tệp đã tải lên.
  + Hoạt động:
    - Người dùng có thể xem chi tiết tệp đa phương tiện đã tải lên trong hệ thống
    - Hệ thống hiển thị ra thông tin về tệp mà người dùng đã tải lên (Kể cả có thay đổi).
* Xem trước thủy vân trước khi nhúng
  + **Mục tiêu:** Người dùng có thể xem trước thủy vân mình đã chọn.
  + Hoạt động:
    - Hệ thống hiển thị thủy vân người dùng đã chọn.
* Nhúng thủy vân
  + **Mục tiêu:** Người dùng có nhúng thành công thủy vân lên tệp đa phương tiện.
  + Hoạt động:
    - Người dùng lưu sau khi chọn thủy vân của mình.
    - Hệ thống hiển thị ra thông báo và hiển thị tệp đa phương tiện đã được nhúng thủy vân sau khi người dùng ấn lưu.
* Xem thông tin cá nhân
  + **Mục tiêu:** Người dùng xem được thông tin cá nhân.
  + Hoạt động:
    - Người dùng có thể xem thông tin cá nhân sau khi ấn vào trang cá nhân.
    - Hệ thống hiển thị thông tin cá nhân của người dùng sau khi người dùng vào trang cá nhân.

### Thông tin các đối tượng cần xử lý

* Người dùng (User)
  + Thư điện tử (email)
  + Tên người dùng (username)
  + Ảnh người dùng (profile\_picture)
  + Mã định danh google (google\_id)
  + Lần đăng nhập (last\_login\_time)
* Tệp đa phương tiện (Media file)
  + Tên tệp (file\_type)
  + Kích thước tệp (file\_size)
  + Đường dẫn tệp (file\_path)
  + Chiều rộng (width)
  + Chiều cao (height)
  + Mô tả (description)
  + Thời gian tạo (created\_at)
  + Thời gian sửa (updated\_at)
  + Tệp đã nhúng thủy vân (file\_watermarked)
  + Tùy chọn thủy vân(watermark\_options)
    - Loại (type)
    - Nội dung(content)
    - Tọa độ x (position\_x)
    - Tọa độ y (position\_y)
    - Độ trong suốt (opacity)
    - Kích cỡ(size)
    - Màu (color)
    - Mã font (font\_id)

### Kiến trúc hệ thống

#### Frontend

* Vite:
  + Vite được sử dụng trong kiến trúc hệ thống frontend như một công cụ build và môi trường phát triển, giúp xây dựng và quản lý mã nguồn của ứng dụng một cách hiệu quả và tối ưu. Trong kiến trúc này, Vite đóng vai trò quan trọng không chỉ trong giai đoạn phát triển mà còn trong việc tối ưu hóa mã nguồn cho môi trường production.
  + Vite xử lý việc cung cấp môi trường phát triển với thời gian phản hồi nhanh, và quá trình build mã nguồn thành phiên bản tối ưu cho production. Nó cung cấp một server phát triển để các lập trình viên có thể xem trước ứng dụng, đồng thời hỗ trợ nhiều tính năng như:
    - Nạp và biên dịch module nhanh chóng.
    - Tối ưu hóa các tệp static như JavaScript, CSS, hình ảnh.
    - Xử lý các thư viện bên ngoài và plugin.
    - Vite đặc biệt hữu ích khi làm việc với các dự án frontend lớn nhờ khả năng tối ưu hóa hiệu suất tốt hơn các công cụ build truyền thống.
* Vue.js:
  + Vue.js là một framework JavaScript tiến bộ (progressive framework) dùng để xây dựng giao diện người dùng (UI). Nó có thể được áp dụng dần dần vào các ứng dụng web, từ những phần nhỏ cho đến toàn bộ ứng dụng phức tạp. Vue.js rất linh hoạt và dễ tích hợp với các dự án hiện có, đặc biệt mạnh mẽ trong việc phát triển các ứng dụng web tương tác và SPA (Single Page Applications).
  + Vue.js chủ yếu xử lý các phần liên quan đến giao diện người dùng và các tương tác phía frontend:
  + Xây dựng giao diện người dùng (UI Components):
    - Chia nhỏ giao diện thành các thành phần (components): Trong Vue.js, giao diện của dự án được chia thành các thành phần nhỏ, dễ quản lý. Ví dụ, trong một ứng dụng thương mại điện tử, bạn có thể có các thành phần như Header, Footer, ProductList, ShoppingCart, v.v. Mỗi thành phần này có thể được tái sử dụng và có logic riêng, giúp mã nguồn dễ đọc và bảo trì.
    - Template: Sử dụng HTML kết hợp với Vue directives (chỉ thị của Vue như v-if, v-for, v-bind,...) để xây dựng giao diện. Vue sẽ tự động cập nhật khi có sự thay đổi trong dữ liệu (reactive updates).
  + Quản lý và cập nhật dữ liệu động (Data Binding):
    - Vue.js giúp đồng bộ dữ liệu giữa giao diện và dữ liệu của ứng dụng thông qua cơ chế two-way data binding. Khi người dùng nhập dữ liệu vào các form hoặc tương tác với giao diện, dữ liệu phía backend có thể tự động cập nhật và ngược lại.
  + Xử lý sự kiện người dùng:
    - Vue.js cung cấp cách thức dễ dàng để xử lý các sự kiện người dùng, như nhấp chuột, nhập liệu, kéo thả, hoặc các sự kiện từ bàn phím.
  + Quản lý trạng thái của ứng dụng:
    - Quản lý trạng thái ứng dụng: Khi dự án trở nên phức tạp với nhiều thành phần giao diện liên quan đến nhau, Vuex (thư viện quản lý trạng thái của Vue.js) sẽ được sử dụng để quản lý trạng thái ứng dụng.
  + Xử lý điều hướng giữa các trang (Routing):
    - Vue.js sử dụng Vue Router để quản lý việc chuyển trang mà không cần tải lại toàn bộ trang (SPA - Single Page Application). Điều này giúp trải nghiệm người dùng mượt mà hơn.
  + Tạo hiệu ứng và hoạt ảnh (Animations):
    - Vue.js hỗ trợ các hiệu ứng chuyển động mượt mà khi thêm, xóa, hoặc cập nhật các thành phần trong giao diện. Bạn có thể dễ dàng thêm các hiệu ứng CSS hoặc sử dụng thư viện animation có sẵn.
* Axios:
  + Axios là một thư viện JavaScript được sử dụng để thực hiện các yêu cầu HTTP từ phía client (trình duyệt) hoặc phía server (Node.js) đến API hoặc các máy chủ web. Nó cung cấp một giao diện đơn giản và mạnh mẽ để gửi các yêu cầu HTTP như GET, POST, PUT, DELETE và xử lý các phản hồi từ máy chủ. Axios thường được sử dụng trong các dự án frontend để tương tác với backend, lấy dữ liệu hoặc gửi dữ liệu lên server.
* Element plus:
  + Element plus là một thư viện giao diện người dùng (UI) dành cho Vue 3, cung cấp các thành phần (components) UI sẵn có và dễ sử dụng. Được xây dựng dựa trên Vue.js, Element Plus giúp các nhà phát triển xây dựng giao diện người dùng một cách nhanh chóng và nhất quán. Thư viện này được phát triển từ Element UI (dành cho Vue 2) và hiện đã được tối ưu hóa để hoạt động tốt trên nền tảng Vue 3.
  + Element Plus đóng vai trò quan trọng trong việc giúp các nhà phát triển xây dựng giao diện người dùng một cách nhanh chóng, tối ưu và dễ bảo trì. Thư viện này không chỉ cung cấp các thành phần giao diện sẵn có, mà còn giúp đảm bảo tính nhất quán, trải nghiệm người dùng tốt, và dễ dàng tích hợp với Vue 3 trong các dự án phát triển ứng dụng web:
  + Tạo giao diện nhanh chóng và nhất quán:
    - Thay vì tự xây dựng các thành phần giao diện từ đầu, Element Plus cung cấp sẵn một bộ các thành phần giao diện, từ đơn giản đến phức tạp, như các nút (buttons), hộp thoại (modals), bảng dữ liệu (tables), và biểu mẫu (forms).
  + Cải thiện hiệu suất và trải nghiệm người dùng:
    - Nhờ vào các thành phần giao diện được tối ưu hóa sẵn, Element Plus giúp cải thiện tốc độ phát triển, đồng thời đảm bảo giao diện trực quan, thân thiện với người dùng.
  + Quản lý layout của trang web:
    - Element Plus cung cấp các thành phần bố cục như Container, Header, Aside, Main, giúp quản lý và sắp xếp bố cục của trang web một cách dễ dàng, phù hợp với các ứng dụng lớn hoặc giao diện phức tạp.

#### Backend

* Lớp Backend đóng vai trò chính trong việc quản lý toàn bộ quá trình xử lý tệp, nhúng thủy vân, lưu trữ tệp, và xác thực người dùng. Hệ thống sử dụng Django Rest Framework (DRF) để xây dựng các API RESTful, giúp kết nối với frontend và xử lý các yêu cầu từ người dùng, đồng thời quản lý thông tin và tệp đa phương tiện thông qua cơ sở dữ liệu MongoDB.
* Backend trong hệ thống này có các nhiệm vụ chính sau:
  + Quản lý xác thực người dùng và phân quyền: Hệ thống sử dụng Google OAuth để xác thực người dùng, đảm bảo rằng chỉ những người dùng đã xác thực mới có quyền truy cập các dịch vụ.
  + Xử lý nhúng thủy vân: Backend nhận các tệp từ người dùng và thực hiện việc nhúng thủy vân số lên các tệp đa phương tiện (hình ảnh, video, PDF). Toàn bộ quá trình này được thực hiện bằng cách sử dụng các thư viện Python chuyên biệt cho từng loại tệp.
  + Lưu trữ và quản lý tệp: Sau khi tệp đã được xử lý, backend lưu trữ tệp và cung cấp cho người dùng khả năng tải tệp xuống hoặc quản lý thông tin của tệp đó.
  + Tích hợp với cơ sở dữ liệu NoSQL (MongoDB): Backend lưu trữ thông tin về người dùng, các tệp đã tải lên và tình trạng xử lý nhúng thủy vân của chúng trong MongoDB.
* Các thành phần chính trong Backend:

**Django Rest Framework (DRF)**

* Django Rest Framework (DRF) là nền tảng chính của hệ thống backend, giúp xây dựng các API RESTful. DRF được sử dụng để cung cấp các API cho việc:
  + Tải lên tệp: Người dùng có thể tải các tệp hình ảnh, video và PDF lên hệ thống thông qua các endpoint của DRF. Mỗi tệp được lưu trữ tạm thời trước khi được xử lý.
  + Nhúng thủy vân: Khi nhận được yêu cầu từ frontend, DRF sẽ tiếp nhận các thông tin về thủy vân (nội dung, vị trí, kích thước, v.v.), sau đó truyền chúng đến các hàm xử lý thủy vân.
  + Quản lý người dùng: DRF tích hợp với Google OAuth để xác thực người dùng, đảm bảo tính bảo mật và phân quyền cho từng người dùng.
  + Xử lý tệp và trả về kết quả: Sau khi hoàn thành việc nhúng thủy vân, backend trả lại tệp đã xử lý cho người dùng qua API.
* Cấu trúc API của DRF giúp chia nhỏ các chức năng của backend thành các endpoint rõ ràng và dễ sử dụng. Các API endpoint chính bao gồm:
  + /upload: Tải tệp lên hệ thống.
  + /watermark: Nhúng thủy vân vào tệp.
  + /files: Xem và quản lý các tệp đã tải lên.
  + /auth: Xác thực và quản lý phiên đăng nhập của người dùng.

**Xử lý nhúng thủy vân**

* Backend sử dụng các thư viện Python để xử lý nhúng thủy vân cho các loại tệp khác nhau:
  + Pillow: Thư viện này được sử dụng cho việc nhúng thủy vân vào hình ảnh. Pillow hỗ trợ các tính năng như vẽ văn bản hoặc hình ảnh thủy vân lên trên các tệp JPEG và PNG, đồng thời có thể tùy chỉnh vị trí, kích thước và độ trong suốt của thủy vân.
  + MoviePy: Đối với tệp video, MoviePy được sử dụng để nhúng thủy vân lên từng khung hình của video. MoviePy xử lý các định dạng phổ biến như MP4, MOV và AVI.
* Mỗi loại tệp (hình ảnh, video) được xử lý riêng biệt thông qua các hàm và dịch vụ khác nhau trong backend. Hệ thống đảm bảo rằng quá trình nhúng thủy vân không làm giảm chất lượng của tệp gốc, đồng thời giữ cho trải nghiệm người dùng được liền mạch.

**Lưu trữ tệp và xử lý tạm thời**

* Backend sử dụng Django File System Storage để quản lý lưu trữ tệp. Sau khi người dùng tải tệp lên, hệ thống sẽ lưu trữ chúng trên VPS trong quá trình xử lý. Các tệp này được lưu trong thư mục tạm thời trước khi nhúng thủy vân hoàn tất, và sau đó sẽ được lưu trữ chính thức hoặc trả về cho người dùng.
  + Lưu trữ tạm thời: Tệp được tải lên và lưu trữ trên hệ thống trước khi xử lý. Sau khi hoàn thành quá trình nhúng thủy vân, tệp đã được xử lý sẽ được lưu lại trong hệ thống để người dùng có thể tải về sau này.

**Cơ sở dữ liệu NoSQL (MongoDB)**

* MongoDB đóng vai trò là cơ sở dữ liệu chính cho backend, lưu trữ thông tin liên quan đến người dùng, tệp đã tải lên, và trạng thái xử lý của chúng. Với cấu trúc không cần bảng (schema-less), MongoDB phù hợp để lưu trữ các dữ liệu không có cấu trúc như tệp đa phương tiện.
  + Thông tin người dùng: MongoDB lưu trữ thông tin cơ bản của người dùng (như tên, email) và các dữ liệu liên quan đến quyền truy cập.
  + Thông tin tệp: Mỗi tệp được lưu trữ cùng với các thông tin như loại tệp, kích thước, thời gian tải lên, và trạng thái đã được nhúng thủy vân hay chưa. MongoDB giúp truy xuất và quản lý tệp dễ dàng khi số lượng tệp tăng lên.
  + Lịch sử xử lý: Cơ sở dữ liệu cũng ghi lại các hành động của người dùng liên quan đến việc tải lên và nhúng thủy vân, giúp theo dõi và quản lý quá trình xử lý.

### Quy trình hoạt động của hệ thống

* Hệ thống hoạt động dựa trên quy trình logic từ việc tải lên tệp cho đến nhúng thủy vân và trả về kết quả cho người dùng. Quy trình được thiết kế đảm bảo tính mạch lạc, giúp người dùng dễ dàng thao tác và nhận kết quả nhanh chóng.
  + Tải lên tệp: Người dùng đầu tiên chọn tệp đa phương tiện từ thiết bị cá nhân để tải lên hệ thống thông qua giao diện người dùng. Sau khi chọn tệp, giao diện frontend sẽ gửi yêu cầu HTTP đến API backend thông qua Axios. Tệp sẽ được lưu trữ tạm thời trên VPS để xử lý.
  + Nhúng thủy vân: Sau khi tệp được tải lên thành công, người dùng có thể chọn các tùy chọn nhúng thủy vân, bao gồm loại thủy vân (văn bản hoặc hình ảnh), vị trí, kích thước, màu sắc, và độ trong suốt. Hệ thống backend sẽ sử dụng các thư viện như Pillow (cho hình ảnh), MoviePy (cho video), hoặc ReportLab (cho PDF) để thực hiện quá trình nhúng thủy vân vào tệp.
  + Xem trước: Trước khi xác nhận nhúng thủy vân, hệ thống cung cấp chức năng xem trước để người dùng có thể kiểm tra kết quả hiển thị. Điều này giúp người dùng chắc chắn rằng thủy vân đã được nhúng đúng vị trí, với các tùy chỉnh phù hợp trước khi lưu kết quả cuối cùng.
  + Lưu tệp đã nhúng thủy vân: Sau khi người dùng xác nhận, tệp đã nhúng thủy vân sẽ được lưu trữ trên Cloud Server và có thể được tải về từ hệ thống dưới dạng một tệp mới. Tệp gốc vẫn được giữ nguyên, đảm bảo rằng người dùng có thể nhúng lại thủy vân nếu cần.
* Quản lý tệp: Sau khi tệp được nhúng thủy vân, người dùng có thể truy cập trang quản lý tệp để xem lại tệp đã nhúng, chỉnh sửa hoặc xóa nếu cần thiết. Mỗi tệp sẽ hiển thị thông tin liên quan như tên, kích thước, thời gian tải lên và trạng thái nhúng thủy vân.

### Cấu trúc và quy trình nhúng thủy vân

* **Nhúng thủy vân vào hình ảnh**

**Bước 1: Nhận dữ liệu đầu vào**

* + Loại thủy vân: Người dùng có thể chọn nhúng thủy vân dạng văn bản hoặc hình ảnh.
  + Nội dung và vị trí: Nếu thủy vân là văn bản, hệ thống sẽ nhận nội dung văn bản và tọa độ (X, Y) nơi văn bản sẽ được nhúng trên hình ảnh.
  + Kích thước, màu sắc, và độ mờ: Kích thước văn bản (font size), màu sắc (hex) và độ mờ (opacity) cũng được chỉ định bởi người dùng. Những thông số này sẽ quyết định giao diện của thủy vân trên hình ảnh.
  + Font chữ: Font chữ cho văn bản cũng được tải từ cơ sở dữ liệu.

**Bước 2: Tạo thủy vân và xử lý hình ảnh**

* + Tải hình ảnh gốc: Sử dụng thư viện PIL (Pillow) để mở hình ảnh cần được nhúng thủy vân. Hình ảnh được mở dưới dạng đối tượng RGBA (hỗ trợ độ trong suốt).
  + Tạo lớp thủy vân: Một lớp mới (dạng hình ảnh) được tạo trên ảnh gốc để vẽ thủy vân. Với thủy vân dạng văn bản:
    - Sử dụng ImageDraw.Draw để vẽ văn bản lên lớp này tại vị trí (X, Y).
    - Font chữ, kích thước, màu sắc và độ mờ được áp dụng theo yêu cầu đầu vào.
  + Kết hợp lớp thủy vân và ảnh gốc: Thông qua quá trình alpha-composite (trộn kênh alpha), lớp thủy vân được gắn vào hình ảnh gốc. Quá trình này giữ nguyên độ trong suốt của thủy vân, tạo hiệu ứng mờ phù hợp trên ảnh.

**Bước 3: Lưu và trả kết quả**

* + Hình ảnh với thủy vân sau khi xử lý sẽ được lưu dưới dạng file mới trên máy chủ. URL của file đã được lưu sau đó sẽ được trả về cho người dùng.
* **Nhúng thủy vân vào video**

**Bước 1: Nhận dữ liệu đầu vào**

* + Loại thủy vân: Người dùng có thể chọn thủy vân dạng văn bản hoặc hình ảnh.
  + Nội dung, vị trí, kích thước: Với thủy vân dạng văn bản, hệ thống nhận nội dung văn bản, vị trí (tọa độ X, Y), kích thước font, màu sắc và độ mờ.
  + Thời gian: Thủy vân sẽ xuất hiện trong suốt chiều dài video.

**Bước 2: Xử lý video và nhúng thủy vân**

* + Tải video: Sử dụng thư viện moviepy.editor, hệ thống tải tệp video cần xử lý.
  + Tạo thủy vân dạng văn bản:
    - Nếu thủy vân là văn bản, một hình ảnh tạm thời (dạng PNG) được tạo từ văn bản, sử dụng thư viện Pillow. Văn bản được vẽ lên hình ảnh PNG này với các thuộc tính như font chữ, màu sắc, và độ mờ.
    - Hình ảnh này sau đó được chuyển thành đối tượng ImageClip của moviepy để dễ dàng ghép vào video.
  + Tạo thủy vân dạng hình ảnh:
    - Nếu thủy vân là hình ảnh, tệp hình ảnh thủy vân được tải lên thông qua request của người dùng. Hệ thống lưu trữ tạm thời hình ảnh này trên máy chủ, và sau đó tạo đối tượng ImageClip từ hình ảnh này.
  + Kết hợp thủy vân và video:
    - Thủy vân (dạng văn bản hoặc hình ảnh) được ghép vào video bằng cách sử dụng CompositeVideoClip. Đây là quá trình kết hợp video gốc và lớp thủy vân dưới dạng các lớp khác nhau trên cùng một khung hình.
    - Hệ thống đặt thủy vân tại vị trí (X, Y) do người dùng chỉ định và áp dụng độ mờ (opacity).

**Bước 3: Lưu và trả kết quả**

* + Video đã được nhúng thủy vân được lưu lại dưới dạng tệp mới. Hệ thống sử dụng codec libx264 để nén video và lưu trữ nó trên máy chủ.
  + URL của video mới sau đó được trả về cho người dùng để tải xuống.

### Thử nghiệm hệ thống trên các sản phẩm đa phương tiện

* Trong quá trình nghiên cứu, hệ thống nhúng thủy vân số đã được thử nghiệm trên nhiều loại sản phẩm đa phương tiện khác nhau như hình ảnh, video. Kết quả thử nghiệm được đánh giá theo các tiêu chí về độ bền vững của thủy vân, tính bảo mật, chất lượng nội dung và khả năng sử dụng.
* Thử nghiệm trên hình ảnh:
  + Quy trình: Hệ thống nhận hình ảnh ở các định dạng phổ biến như JPEG và PNG. Người dùng có thể nhúng thủy vân dưới dạng văn bản hoặc hình ảnh, tùy chỉnh các thông số như vị trí, kích thước, màu sắc và độ trong suốt.
  + Kết quả:
    - Thủy vân được nhúng vào hình ảnh một cách chính xác tại các vị trí đã định, với độ mờ phù hợp và không làm ảnh hưởng đến chất lượng hình ảnh gốc.
* Thử nghiệm trên video:
  + Quy trình: Hệ thống hỗ trợ nhúng thủy vân số vào video với các định dạng phổ biến như MP4 và MOV. Thủy vân có thể là hình ảnh hoặc văn bản, được ghép trực tiếp lên các khung hình của video.
  + Kết quả:
    - Thủy vân hiển thị ổn định trên toàn bộ chiều dài video mà không làm giảm chất lượng khung hình.
    - Hệ thống xử lý nhúng thủy vân vào các video ngắn có độ phân giải từ HD đến Full HD với thời gian xử lý hợp lý. Tuy nhiên, với các video có độ phân giải cực cao hoặc thời lượng dài, thời gian xử lý có thể tăng đáng kể.

### Hình ảnh ứng dụng

#### Trang chủ

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

#### Trang hồ sơ

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

#### Trang ảnh

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

#### Trang video

A screenshot of a video player

Description automatically generated

A screenshot of a video player

Description automatically generated

Trang API

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## ❖ CHƯƠNG 4: KẾT LUẬN

### 4.1 Yêu cầu đặt ra:

* Đề tài nghiên cứu này đặt mục tiêu xây dựng và thử nghiệm hệ thống nhúng thủy vân số để bảo vệ bản quyền các sản phẩm đa phương tiện, thông qua việc sử dụng ngôn ngữ lập trình Python và các thư viện hỗ trợ. Hệ thống phải đáp ứng được yêu cầu tích hợp nhúng thủy vân số vào các loại dữ liệu như hình ảnh, video, đồng thời đảm bảo tính khả dụng, hiệu quả và dễ sử dụng cho người dùng.

### 4.2 Kết quả đã thực hiện nghiên cứu:

* Trong quá trình thực hiện đề tài, nhóm nghiên cứu đã đạt được các kết quả sau:
* Phân tích hiện trạng: Đã khảo sát và đánh giá các kỹ thuật nhúng thủy vân số hiện có, đặc biệt là khi sử dụng Python với các thư viện như Pillow, MoviePy và ReportLab. Qua đó, xác định những ưu điểm của Python trong việc xử lý đa phương tiện và bảo vệ bản quyền.
* Phát triển hệ thống nhúng thủy vân số: Nhóm đã xây dựng thành công hệ thống ứng dụng web với các chức năng chính bao gồm tải lên tệp đa phương tiện, nhúng thủy vân (dưới dạng văn bản hoặc hình ảnh), và cung cấp tính năng xem trước trước khi hoàn tất việc nhúng. Hệ thống cũng hỗ trợ các tính năng quản lý tệp và xác thực người dùng.
* Thử nghiệm và đánh giá: Hệ thống đã được thử nghiệm trên nhiều loại dữ liệu khác nhau, bao gồm hình ảnh, video. Kết quả cho thấy hệ thống hoạt động hiệu quả trong việc bảo vệ bản quyền sản phẩm mà không làm ảnh hưởng đáng kể đến chất lượng nội dung gốc. Ngoài ra, quá trình thử nghiệm cũng giúp nhóm nhận diện một số hạn chế liên quan đến hiệu suất khi xử lý các tệp có kích thước lớn.

### 4.3 Hướng phát triển tiếp:

* Dựa trên kết quả nghiên cứu và thử nghiệm, nhóm đề xuất một số hướng phát triển tiếp theo để cải thiện hệ thống:
  + Tối ưu hóa hiệu suất: Tăng cường khả năng xử lý các tệp đa phương tiện có kích thước lớn hơn, đồng thời cải thiện thời gian phản hồi của hệ thống khi nhúng thủy vân vào video có độ phân giải cao.
  + Bổ sung tính năng bảo mật: Áp dụng các kỹ thuật mã hóa để tăng cường tính bảo mật của thủy vân số, đảm bảo thủy vân không thể dễ dàng bị loại bỏ hoặc chỉnh sửa.
  + Phát triển thêm các tính năng nâng cao: Cung cấp tùy chọn nhúng thủy vân số ẩn, giúp bảo vệ bản quyền mà không ảnh hưởng đến trải nghiệm người dùng. Đồng thời, mở rộng hệ thống để có thể quản lý được nhiều định dạng tệp đa phương tiện hơn.
* Triển khai hệ thống trên các môi trường khác nhau: Đưa hệ thống vào thử nghiệm trên các nền tảng khác nhau như điện toán đám mây hoặc các hệ thống phi tập trung để tăng tính khả dụng và mở rộng quy mô.