**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐÀO TẠO LIÊN TỤC**

**BÁO CÁO THỰC TẬP KỸ THUẬT**

**Xây dựng phần mềm số hoá tài liệu**

|  |  |
| --- | --- |
| *Giáo viên hướng dẫn:* | **ThS. Lê Quốc Trung** |
| *Sinh viên thực hiện:* | **Nguyễn Phương Thuỳ**  **Phạm Thành Vân**  **Lê Thị Hoàng Yến** |

**Hà Nội, 11/2020**

MỤC LỤC

[DANH MỤC HÌNH ẢNH 3](#_Toc56427778)

[DANH MỤC BẢNG 4](#_Toc56427779)

[I. GIỚI THIỆU 5](#_Toc56427780)

[1. Ngôn ngữ lập trình C# 5](#_Toc56427781)

[1.1. Đặc điểm của ngôn ngữ 5](#_Toc56427782)

[1.2. Đặc trưng của ngôn ngữ C# 5](#_Toc56427783)

[2. HTML 7](#_Toc56427784)

[2.1. Giới thiệu 7](#_Toc56427785)

[2.2. Đánh dấu 8](#_Toc56427786)

[3. CSS 8](#_Toc56427787)

[4. JavaScript 9](#_Toc56427788)

[4.1. Giới thiệu 9](#_Toc56427789)

[4.2. Java, JavaScript và Jscript 9](#_Toc56427790)

[5. ReactJS 10](#_Toc56427791)

[6. Tesseract 12](#_Toc56427792)

[II. Phần mềm số hoá tài liệu 13](#_Toc56427793)

[1. Kiến trúc hệ thống 13](#_Toc56427794)

[2. Thiết kế cơ sở dữ liệu 14](#_Toc56427795)

[3. Hoạt động số hóa tài liệu của hệ thống 16](#_Toc56427796)

[4. Sử dụng phần mềm 16](#_Toc56427797)

# DANH MỤC HÌNH ẢNH

[Hình 1. Cấu trúc 1 trang HTML và hiển thị trên trình duyệt 7](#_Toc56427005)

[Hình 2. Trang HTML có sử dụng CSS 9](#_Toc56427006)

[Hình 3. Virtual DOM 10](#_Toc56427007)

[Hình 4. One-way data binding 11](#_Toc56427008)

[Hình 5. Kiến trúc hệ thống 13](#_Toc56427009)

[Hình 6. Các bảng và quan hệ giữa các bảng trong CSDL 14](#_Toc56427010)

[Hình 7. Màn hình đăng ký tài khoản 16](#_Toc56427011)

[Hình 8. Màn hình đăng nhập 17](#_Toc56427012)

[Hình 9. Giao diện chính của phần mềm 17](#_Toc56427013)

# DANH MỤC BẢNG

[Bảng 1. Bảng User 14](#_Toc56427039)

[Bảng 2. Bảng Job 15](#_Toc56427040)

[Bảng 3. Bảng FileStorage 16](#_Toc56427041)

# GIỚI THIỆU

## Ngôn ngữ lập trình C#

C# (C Sharp) là một ngôn ngữ lập trình hướng đối tượng đa năng vô cùng mạnh mẽ được phát triển bởi Microsoft, C# là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

C# được thiết kế chủ yếu bởi Anders Hejlsberg kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC. Phiên bản gần đây nhất là 9.0, được phát hành vào năm 2020 cùng với Visual Studio 2019 phiên bản 16.8.

### Đặc điểm của ngôn ngữ

C#, theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến .NET Framework mà tất cả các chương trình.NET chạy, và nó phụ thuộc mạnh mẽ vào framework này. Mọi dữ liệu cơ sở đều là đối tượng, được cấp phát và hủy bỏ bởi trình dọn rác Garbage-Collector (GC), và nhiều kiểu trừu tượng khác chẳng hạn như class, delegate, interface, exception... phản ánh rõ ràng những đặc trưng của.NET runtime.

So sánh với C và C++, ngôn ngữ này bị giới hạn và được nâng cao ở một vài đặc điểm nào đó, nhưng không bao gồm các giới hạn sau đây:

* Các con trỏ chỉ có thể được sử dụng trong chế độ không an toàn. Hầu hết các đối tượng được tham chiếu an toàn, và các phép tính đều được kiểm tra tràn bộ đệm. Các con trỏ chỉ được sử dụng để gọi các loại kiểu giá trị; còn những đối tượng thuộc bộ gom rác (garbage-collector) thì chỉ được gọi bằng cách tham chiếu.
* Các đối tượng không thể được giải phóng tường minh.
* Chỉ có đơn kế thừa, nhưng có thể cài đặt nhiều interface trừu tượng (abstract interfaces). Chức năng này làm đơn giản hóa sự thực thi của thời gian thực thi.
* C# thì an-toàn-kiểu (typesafe) hơn C++.
* Cú pháp khai báo mảng khác nhau("int[] a = new int[5]" thay vì "int a[5]").
* Kiểu thứ tự được thay thế bằng tên miền không gian (namespace).
* C# không có tiêu bản.
* Có thêm Properties, các phương pháp có thể gọi các Properties để truy cập dữ liệu.
* Có reflection.

### Đặc trưng của ngôn ngữ C#

* C# là ngôn ngữ đơn giản, mạnh mẽ
  + C# được dựng trên nền tảng C++ và Java, ảnh hưởng bởi Delphi, VisualBasic nên ngôn ngữ C# được thừa hưởng các ưu điểm vào loại bỏ các yếu điểm của các ngôn ngữ trên, vì vậy nó khá đơn giản, đồng thời loại bỏ các cú pháp dư thừa và thêm vào đó các cú pháp cải tiến hơn
  + C# là ngôn ngữ lâp trình bậc cao, đa nền tảng vì vậy nó dễ dàng tiếp cận và phù hợp cho người mới bắt đầu học, ví dụ câu lệnh kinh điển dành cho người mới băt đầu hoc là in ra dòng chữ "Hello world", với C# ta chỉ cẩn 1 câu lệnh: System.Console.WriteLine("Hello world");
* C# là ngôn ngữ đa năng và hiện đại
  + C# phù hợp cho việc phát triển trong thời đại 4.0, bao gồm việc phát triển web, mobile app, game, học máy và trí tuệ nhân tạo, phát triển đám mây, IoT, blockchain, microservices...
* C# là một ngôn ngữ lập trình hướng đối tượng đồng thời hỗ trợ lâp trình chức năng
  + C# hỗ trợ mạnh mẽ cho phương pháp lâp trình hướng đối tượng, ngoài ra C# còn hỗ trợ các phuơng pháp lập trình chức năng thông qua các biểu thức lamba, khớp mẫu, functions, các thuộc tính bất biến.
* C# là ngôn ngữ gõ tĩnh, định kiểu mạnh, hỗ trợ gõ động.
  + C# được gõ tĩnh nên nó mang đầy đủ các ưu việt của phuơng pháp gõ tĩnh như bảo đảm an toàn kiểu, tự động phân tích và nhận biết lỗi cú pháp ngay trong quá trình viết mã...
  + Ngoài ra khi sử dụng C# kết hợp với IDE Visual Studio, C# được hỗ trợ gợi ý code bởi Visual Studio IntelliCode sử dụng trí tuệ nhân tạo giúp cho việc viết code trở nên nhanh chóng và dễ dàng hơn
* C# là một ngôn ngữ ít từ khóa
  + C# có khoảng hơn 80 từ khóa
* C# là một trong các ngôn ngữ lập trình phổ biến và phát triển nhất
  + Theo TIOBE Index, tính đến tháng 10/2020, C# là ngôn ngữ phổ biến thứ 5 thế giới.
  + Theo PYPL, tính đến tháng 10/2020, C# là ngôn ngữ được cộng đồng quan tâm và chia sẻ nhiều thứ 4 thế giới.
* C# kết hợp chặt chẽ với nền tảng .NET- một khung nền tảng rất mạnh của Microsoft.
* Ngoài ra C# còn có những ưu điểm:
  + C# là ngôn ngữ lập trình mã nguồn mở,vì vậy C# là miễn phí với tất cả mọi người, đồng thời mọi người đều có thể cùng tham gia phát triển, đề xuất thiết kế ngôn ngữ C#
  + C# là ngôn ngữ đa nền tảng vì vậy có thể biên dịch trên nhiều nền tảng máy tính khác nhau (Windows, Linux, MacOS)
  + C# có hiệu suất cao và tốc độ thực thi nhanh do sử dụng trình biên dich trung gian (CLR), điểm cộng nữa là tốc độ phát triển phần mềm nhanh chóng so với đa số các ngôn ngữ hiện tại.
  + C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.
  + C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, song cũng được bổ sung các yếu tố mang tính hiện đại nên dễ dàng tiếp cận cho người mới học và học nhanh với C#.
  + C# có cộng đồng nhà phát triển vô cùng lớn mạnh.
  + C# được phát triển và cải tiến không ngừng với tần suất 1 phiên bản/ 1 năm, đáp ứng các mong muốn cải thiện, cải tiến cho phù hợp với nhu cầu công nghệ của các nhà phát triển.
  + C# có tài liệu tham khảo và hướng dẫn vô cùng phong phú và chất lượng, đồng thời có các buổi hội thảo giới thiệu tính năng mới và định hướng phát triển ngôn ngữ trong tương lai.
  + C# và .NET được đánh giá là có design tốt, vì vậy cú pháp và logic rất nhất quán, mã nguồn C# dễ đọc và mở rộng.
  + C# được thiết kế và phát triển bởi Microsoft nên rất được Microsoft quan tâm và hỗ trợ.

## HTML

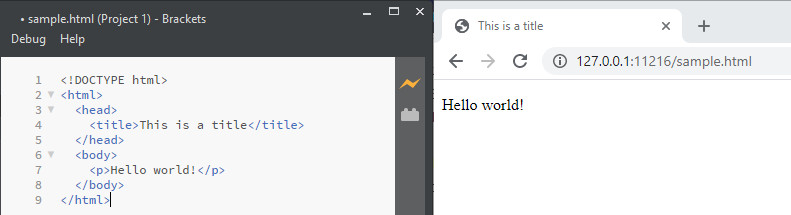
### Giới thiệu

HTML viết tắt cho HyperText Markup Language (Ngôn ngữ đánh dấu siêu văn bản) là ngôn ngữ đánh dấu tiêu chuẩn trong việc thiết kế tài liệu để hiển thị trên trình duyệt. Cùng với CSS và Javascript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web.

HTML được sáng tạo bởi Tim Berners-Lee, nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. HTML đã trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì.

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi dấu “<” và “>” (ví dụ: <html>) và thường là sẽ được khai báo thành 1 cặp, bao gồm thẻ mở và thẻ đóng (ví dụ: <body> và </body>). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví du: <strong>Chữ in đậm</strong>). Một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ thẻ <img>).

Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đôi mở rộng là .html hoặc .htm. Khi một tập tin HTML được hình thành, việc xử lý nó sẽ do trình duyệt đảm nhận. Trình duyệt sẽ đóng vai trò đọc hiểu nội dung HTML từ các thẻ bên trong và hiển thị (render) cho người dùng.



Hình 1. Cấu trúc 1 trang HTML và hiển thị trên trình duyệt

### Đánh dấu

Phần tử HTML hay còn được biết đến với tên Tag hay Entity hay thẻ. HTML là ngôn ngữ đánh dấu, do đó có thể hiểu một phần tử HTML chính là một đoạn văn bản được đánh dấu để thể hiện theo một cách nào đó.

Có bốn loại phần tử đánh dấu trong HTML:

* Đánh dấu *Có cấu trúc* miêu tả mục đích của phần văn bản
* Đánh dấu *trình bày* miêu tả phần hiện hình trực quan của phần văn bản bất kể chức năng của nó là gì (ví dụ: <b>boldface</b> sẽ hiển thị đoạn văn bản **boldface**). Cách dùng đánh dấu trình bày này bây giờ không còn được khuyên dùng mà nó được thay thế bằng cách dùng CSS.
* Đánh dấu *liên kết ngoài* chứa phần liên kết từ trang này đến trang kia (ví dụ: <a href=”<https://www.wikipedia.org/>”>Wikipedia</a>)
* Các thành phần điều khiển giúp tạo ra các đối tượng (ví dụ: các nút và các danh sách)

## CSS

CSS (Cascading Style Sheet) là ngôn ngữ tạo phong cách cho trang web. Nó dùng để tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như là HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người viết web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc, và font chữ.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản. HTML không được thiết kế để gắn tag để giúp định dạng trang web. Bạn chỉ có thể dùng nó để “đánh dấu” lên site.

Những tag như <font> được ra mắt trong HTML phiên bản 3.2, nó gây rất nhiều rắc rối cho lập trình viên. Vì website có nhiều font khác nhau, màu nền và phong cách khác nhau. Để viết lại code cho trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo bởi W3C là để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ không chỉ là một trang chứa văn bản mà không có gì khác.

CSS có cú pháp đơn giản, sử dụng một số từ khóa tiếng Anh để làm tên cho các thuộc tính.



Hình 2. Trang HTML có sử dụng CSS

## JavaScript

### Giới thiệu

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web (phía người dùng) cũng như phía máy chủ (với Nodejs). Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.

Phiên bản mới nhất của JavaScript là ECMAScript 7[3]. ECMAScript là phiên bản chuẩn hóa của JavaScript. Trình duyệt Mozilla phiên bản 1.8 beta 1 có hỗ trợ không đầy đủ cho E4X - phần mở rộng cho JavaScript hỗ trợ làm việc với XML, được chuẩn hóa trong ECMA-357.

### Java, JavaScript và Jscript

Cùng thời điểm Netscape bắt đầu sử dụng công nghệ Java trên trình duyệt Netscape, LiveScript đã được đổi tên thành JavaScript để được chú ý hơn bởi ngôn ngữ lập trình Java lúc đó đang được coi là một hiện tượng. JavaScript được bổ sung vào trình duyệt Netscape bắt đầu từ phiên bản 2.0b3 của trình duyệt này vào tháng 12 năm 1995. Trên thực tế, JavaScript không được phát triển dựa từ Java. Do đó JavaScript chỉ dựa trên các cách đặt tên của Java. Java Script gồm 2 mảng là client-server thực hiện lệnh trên máy của end-user và web-server.

Sau thành công của JavaScript, Microsoft bắt đầu phát triển JScript, một ngôn ngữ có cùng ứng dụng và tương thích với JavaScript. JScript được bổ sung vào trình duyệt Internet Explorer bắt đầu từ Internet Explorer phiên bản 3.0 được phát hành tháng 8 năm 1996.

DOM (Document Object Model), một khái niệm thường được nhắc đến với JavaScript trên thực tế không phải là một phần của chuẩn ECMAScript, DOM là một chuẩn riêng biệt có liên quan chặt chẽ với XML.

## ReactJS

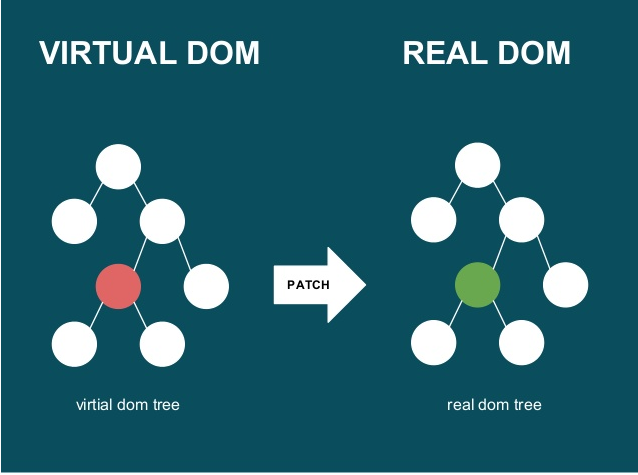
React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javasscript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS.Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và www.instagram.com được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM.

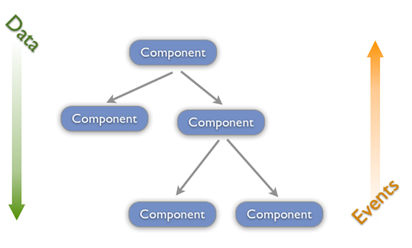
* **Virtual DOM:**

Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này.Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.



Hình 3. Virtual DOM

React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi. Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.



Hình 4. One-way data binding

* **JPX**

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javacsript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng

* **Components**

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

## Tesseract

Tesseract là một OCR (Optical Character Recognition) engine hàng đầu hiện nay. Công cụ này được phân phối với bản quyền mã nguồn mở Apache 2.0. Nó hỗ trợ nhận diện kí tự trên các tập tin hình ảnh và xuất ra dưới dạng kí tự thuần, html, pdf, tsv, invisible-text-only pdf. Người dùng có thể sử dụng trực tiếp hoặc lập trình viên có thể sử dụng các chức năng thông qua API.

Tesseract được phát triển bởi Hewlett-Packard Laboratories Bristol tại Hewleett-Packard Co, Greeley Colorado từ 1985 đến 1994. Sau đó, nó được cập nhật một số thay đổi nhỏ và tạm ngưng phát triển từ sau 1998. Đến năm 2005, Tesseract được phân bố dưới dạng mã nguồn mở bởi HP và được phát triển bởi Google từ năm 2006.

Hiện tại, Tesseract đã phát triển đến version 5.0x và có thể hoạt động trên 3 hệ điều hành phổ biến là Window, Mac và Linux. Công cụ này hỗ trợ nhận diện kí tự của hơn 100 ngôn ngữ khác nhau, bao gồm cả tiếng Việt. Không những thế, chúng ta có thể huấn luyện chương trình dùng Tesseract để có thể nhận diện một ngôn ngữ nào đó. Bên cạnh đó, mã nguồn mở này không hỗ trợ GUI, nên bạn sẽ cần tới ứng dụng của bên thứ ba nếu muốn sử dụng chức năng này.

Đối với các lập trình viên, họ có thể sử dụng các API của Tesseract để xây dựng ứng dụng của mình. Thư viện đó gọi là labtesseract và được cung cấp cho ngôn ngữ C/C++. Trong trường hợp bạn sử dụng ngôn ngữ khác thì cần phải sử dụng các gói hỗ trợ tương ứng

# Phần mềm số hoá tài liệu

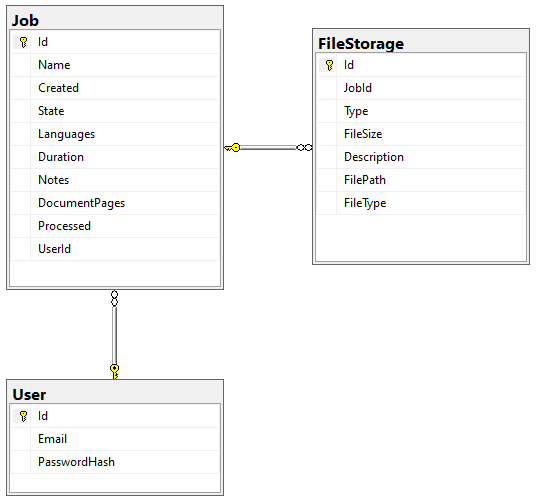
## Kiến trúc hệ thống



Hình 5. Kiến trúc hệ thống

* Front-end
  + Cung cấp giao diện để người dùng tương tác với hệ thống sử dụng trình duyệt.
  + Được xây dựng sử dụng ReactJS.
* Back-end: Gồm 2 thành phần: Application Server và Processing Station
  + Application Server: Cung cấp Web API cho phép tích hợp với các hệ thống khác cho việc thực hiện số hoá tài liệu.
  + Processing Station: Thực hiện nhiệm vụ số hoá tài liệu.
* Database: Cơ sở dữ liệu lưu trữ dữ liệu của hệ thống

## Thiết kế cơ sở dữ liệu



Hình 6. Các bảng và quan hệ giữa các bảng trong CSDL

* **User**: Lưu thông tin người dùng.
* **Job**: Lưu thông tin của tài liệu đưa vào hệ thống.
* **FileStorage**: Lưu thông tin file liên quan đến tài liệu.

**Mô tả chi tiết các bảng trong CSDL**

| **Tên trường** | **Kiểu dữ liệu** | **Not Null?** | **Khóa** | **Mô tả** |
| --- | --- | --- | --- | --- |
| Id | int | Y | PK | Số định danh |
| Email | nvarchar(500) | Y |  | Email của người dùng |
| PasswordHash | Nvarchar(500) | Y |  | Mật khẩu đã được hash |

Bảng 1. Bảng User

| **Tên trường** | **Kiểu dữ liệu** | **Not Null?** | **Khóa** | **Mô tả** |
| --- | --- | --- | --- | --- |
| Id | int | Y | PK | Số định danh |
| Name | nvarchar(255) | Y |  | Tên của tài liệu được đưa vào nhận dạng |
| Created | Datetime2(7) | Y |  | Thời gian tài liệu được đưa vào CSDL |
| State | Int | Y |  | Trạng thái của tài liệu trong quá trình xử lý:  0: đang chờ  1: đang xử lý  2: đã xử lý  -1: lỗi |
| Languages | Nvarchar(250) | Y |  | Các ngôn ngữ sử dụng để nhận dạng |
| Duration | Bigint | N |  | Thời gian nhận dạng |
| Notes | Nvarchar(MAX) | N |  | Ghi chú |
| DocumentPages | int | Y |  | Số trang của tài liệu |
| Processed | Datetime2(7) | N |  | Thời gian hoàn thành xử lý |
| UserId | Int | Y |  | ID của người dùng đưa tài liệu vào xử lý |

Bảng 2. Bảng Job

| **Tên trường** | **Kiểu dữ liệu** | **Not Null?** | **Khóa** | **Mô tả** |
| --- | --- | --- | --- | --- |
| Id | int | Y | PK | Số định danh |
| JobId | Int | Y |  | Job ID của file |
| Type | Int | Y |  | Loại file của tài liệu:  0: file đầu vào  1: file kết quả số hóa |
| FileSize | Bigint | N |  | Kích thước file |
| Description | Nvarchar(50) | N |  | Mô tả loại file |
| FilePath | Nvarchar(255) | N |  | Đường dẫn đến file |
| FileType | Nvarchar(10) | N |  | Phần mở rộng của file |

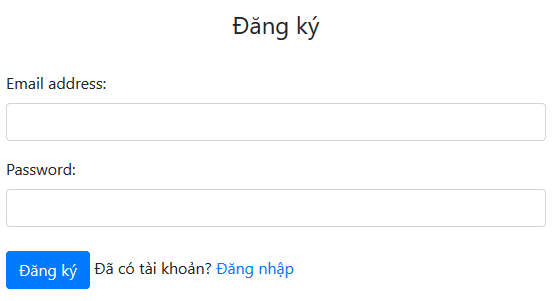
Bảng 3. Bảng FileStorage

## Hoạt động số hóa tài liệu của hệ thống

* **Bước 1:** Front-end gửi file đến Application Server
* **Bước 2:** Application Server lưu thông tin tài liệu vào Database
* **Bước 3:** Processing Station định kỳ gửi yêu cầu đến Application Server để xem có tài liệu nào đang chờ xử lý hay không.
* **Bước 4:** Nếu có tài liệu đang chờ xử lý thì Processing Station gửi yêu cầu đến Application Server để lấy file cần xử lý.
* **Bước 5:** Processing Station thực hiện số hóa tài liệu
* **Bước 6:** Sau khi số hóa tài liệu, Processing Station gửi kết quả đến cho Application Server.
* **Bước 7:** Application Server lưu kết quả số hóa nhận từ Processing Station vào Database.

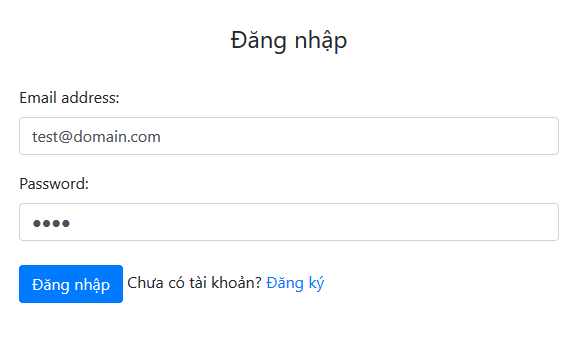
## Sử dụng phần mềm

* **Bước 1:** Đăng ký tài khoản: Người dùng nhập thông tin email, password để đăng ký tài khoản truy cập vào thệ thống

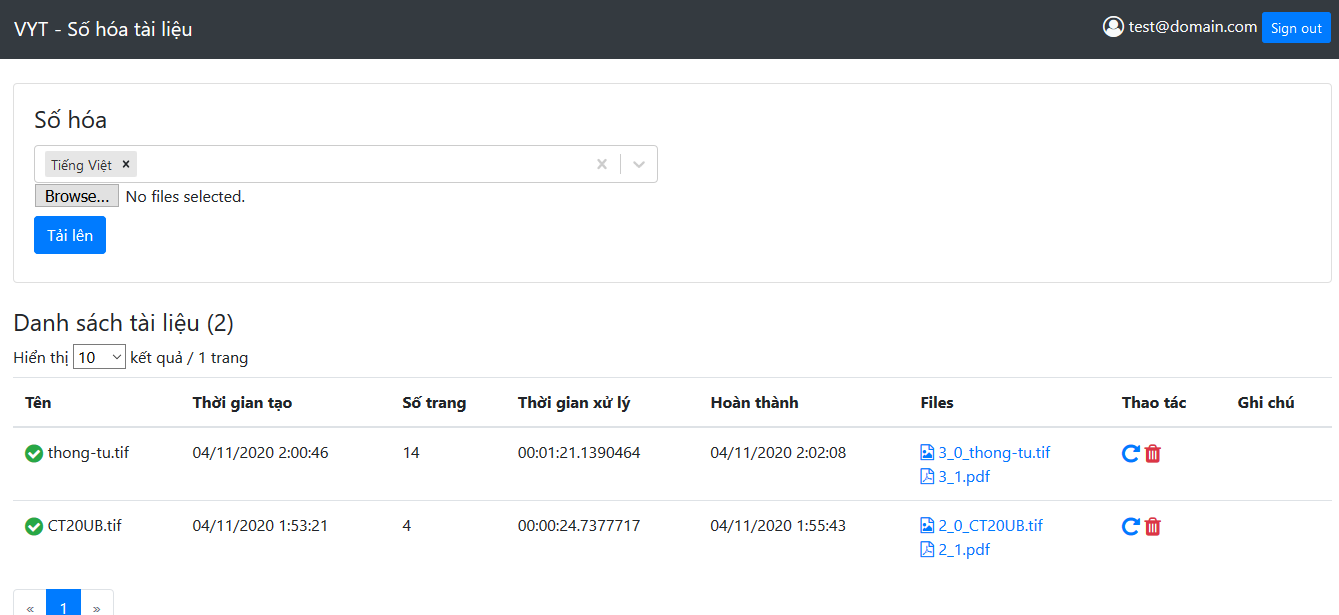


Hình 7. Màn hình đăng ký tài khoản

* **Bước 2:** Đăng nhập hệ thống: Người dùng nhập thông tin email và mật khẩu để đăng nhập hệ thống



Hình 8. Màn hình đăng nhập



Hình 9. Giao diện chính của phần mềm

* **Bước 3:** Tải lên tài liệu cần số hóa
  + Bấm vao nút **Browse...** trong phần **Số hóa**
  + Chọn file ảnh cần số hóa (Có thể chọn nhiều file)
  + Chọn ngôn ngữ sử dụng để số hóa trong ô **Ngôn ngữ**
  + Bấm nút **Tải lên**
* **Bước 4:** Chờ hệ thống xử lý và tải kết quả. Biểu tượng của tài liệu tương ứng với các trạng thái:

: Chờ xử lý

: Đang xử lý

: Đã xử lý



: Lỗi

**Các thao tác khác**

* Thực hiện xử lý lại tài liệu: Bấm vào nút  ở mục **Thao tác**
* Xóa tài liệu: Bấm vào nút ở mục **Thao tác**