**TIỂU LUẬN CHUYÊN NGÀNH 2019**

**KHẢO SÁT THỰC TRẠNG VÀ CHỨC NĂNG CỦA WEBSITE THI TRẮC NGHIỆM ONLINE**

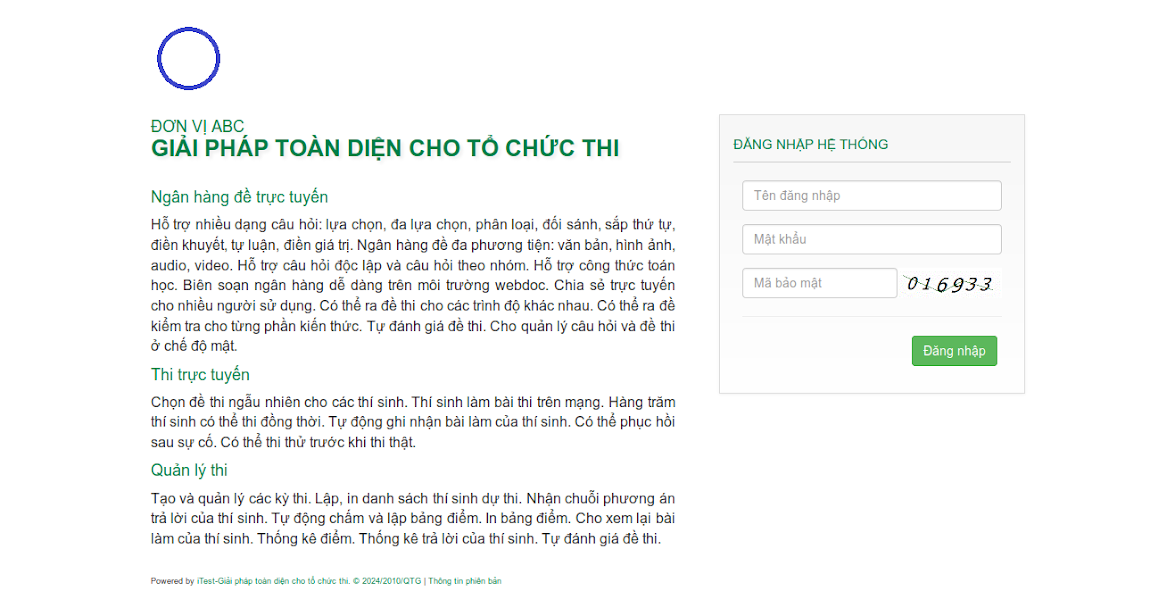
**Thành viên nhóm:**

**Lê Duy Quí-16110190**

**Phan Nguyễn Tấn Tài-16110203**

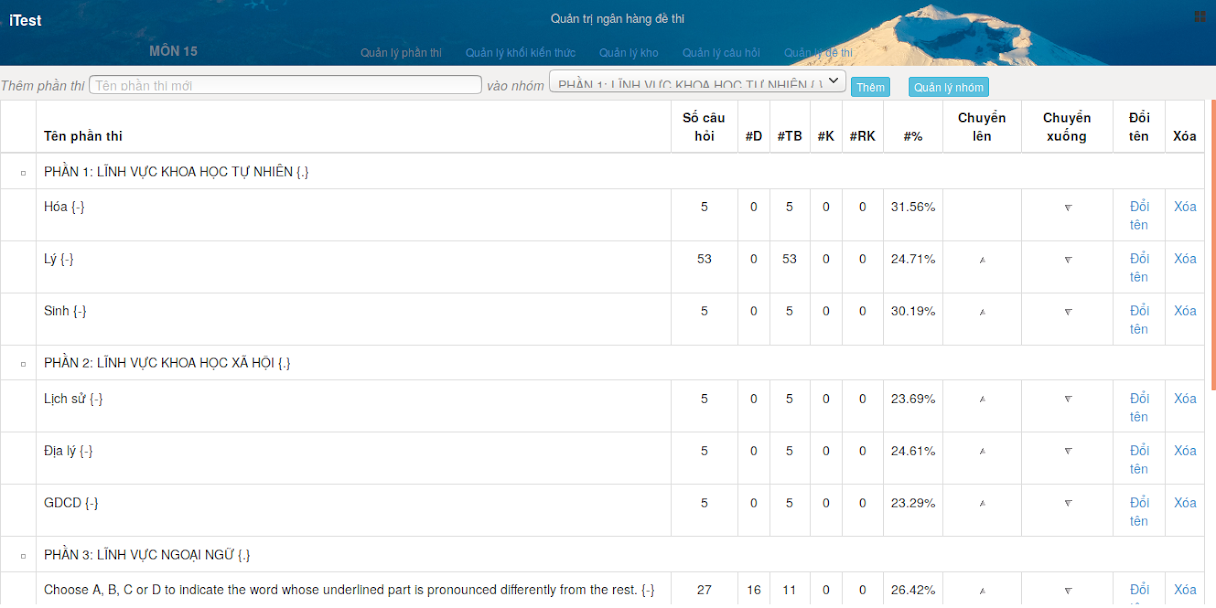
**I-KHẢO SÁT TÍNH NĂNG HỆ THỐNG**

Tính năng của hệ thống được khảo sát thông qua một website mẫu. Các tính năng của website bao gồm:

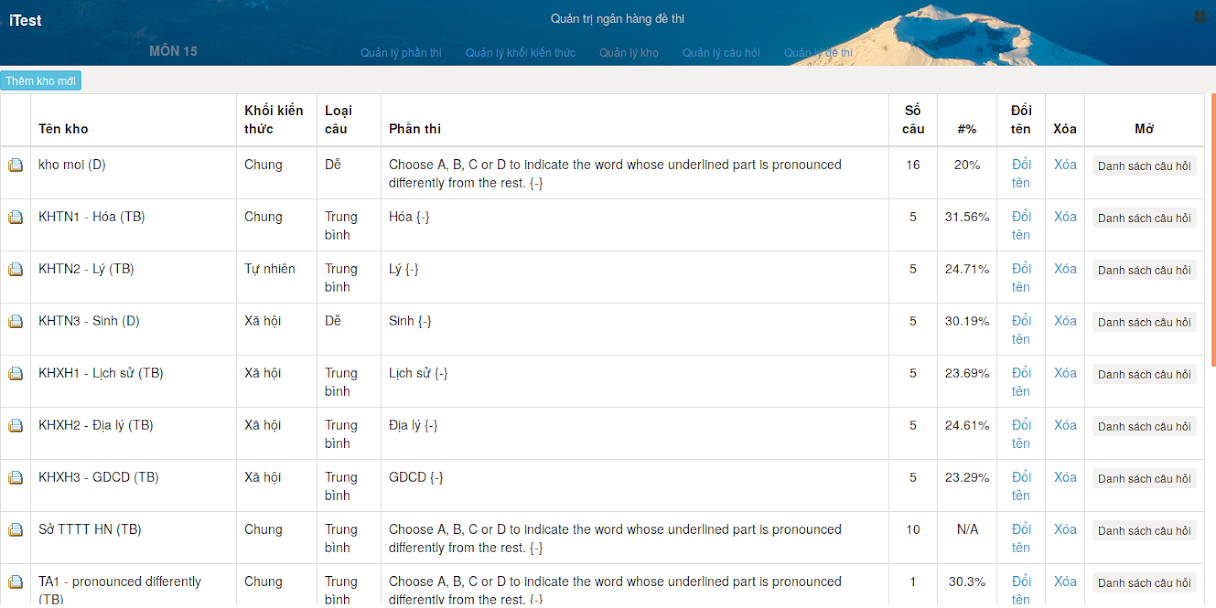


-Thiết kế một trang chủ, ở trang chủ này, chúng ta sẽ có thể giới thiệu kèm về sơ lược về hệ thống cho người dùng tham khảo. Ngoài ra, đăng nhập và đăng kí là những chức năng không thể thiếu ở trang chủ. Với hình ảnh bên trên như một ví dụ minh họa, chúng ta có thể liên kết đăng nhập với các mạng xã hội phổ biến hiện nay nhằm giúp người dùng dễ dàng đăng kí tài khoản và quản lí tài khoản hơn

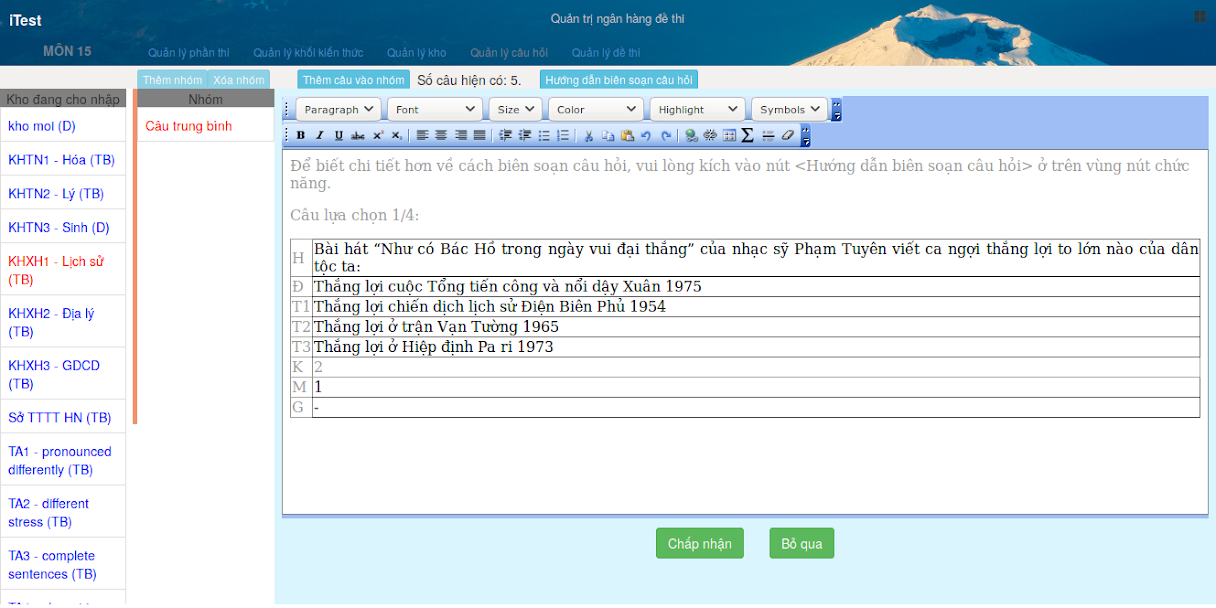
-Sau khi đăng nhập vào hệ thống, người dùng sẽ xuất hiện một giao diện Quản trị ngân hàng đề thi:



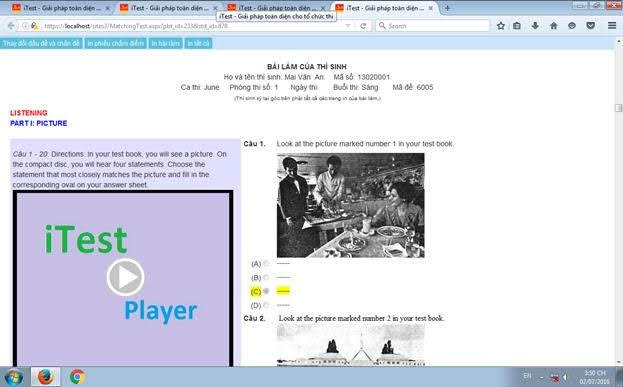
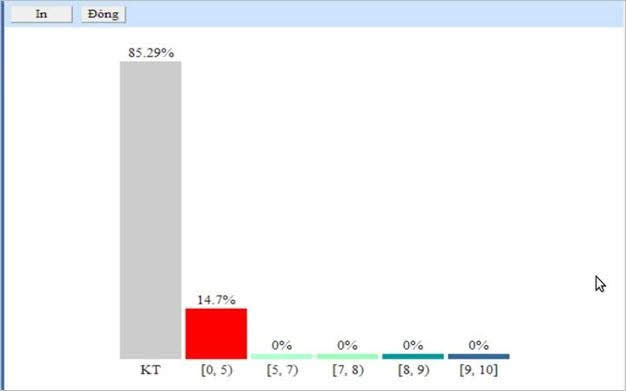
-Với giao diện này, người dùng sẽ tạo được một thư viện đề thi của mình trên nhiều lĩnh vực







-Những hình ảnh phía trên thể hiện tính năng thêm một thư viện câu hỏi vào hệ thống, tại đây chúng ta cũng có thể edit và xóa các câu hỏi đó nếu ta cảm thấy chưa phù hợp

-Những hình ảnh bên dưới thể hiện một trang giao diện sau khi xuất các đề thi ra và được thí sinh làm bài nhấn vào để làm bài. Ngoài ra, hệ thống cũng sẽ thống kê kết quả qua dạng biểu đồ cột.

-Chức năng giám sát phòng thi cũng là một trong những chức năng rất cần thiết cho một hệ thống thi trắc nghiệm online, nhằm giúp cho các thí sinh có thể nghiêm túc hơn trong việc làm bài của mình





-Ngoài ra hệ thống website sẽ được phát triển thêm với những tính năng như: quản lí học viên tham gia thi, quản lí user, thống kê mức độ khó dễ của câu hỏi thông qua những kết quả của học viên,…

**II-PHẠM VI PHÂN BỐ CỦA HỆ THỐNG**

-Hệ thống sẽ được phát triển cho các trường học tổ chức các kì thi cuối khóa, kết thúc học kì với hình thức trắc nghiệm online và chấm điểm tự động sau khi hoàn thành bài thi. Mỗi người dùng sẽ có một user và danh sách các môn học sẽ được bảo mật cho từng user.

**III-CÔNG NGHỆ ÁP DỤNG TRONG HỆ THỐNG**

Các công nghệ được áp dụng trong hệ thống lần này:

-Front End: Angular 8, Bootstrap 6

-Back End: Node Js

-Cơ sở dữ liệu: Mongo db

**1. FRONT END**

**1.1 Bootstrap**

-Bootstrap là 1 framework HTML, CSS, và JavaScript cho phép người dùng dễ dàng thiết kế website theo 1 chuẩn nhất định, tạo các website thân thiện với các thiết bị cầm tay như mobile, ipad, tablet,...

Tại sao lại như vậy?

-Bootstrap bao gồm những cái cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm nhiều Component, Javascript hỗ trợ cho việc thiết kế reponsive của bạn dễ dàng, thuận tiện và nhanh chóng hơn.

***Tại sao chúng ta nên sử dụng Bootstrap?***

-Bootstrap là một trong những framework được sử dụng nhiều nhất trên thế giới để xây dựng nên một website. Bootstrap đã xây dựng nên 1 chuẩn riêng và rất được người dùng ưa chuộng. Chính vì thế, chúng ta hay nghe tới một cụm từ rất thông dụng "Thiết kế theo chuẩn Bootstrap".

Từ cái "chuẩn mực" này, chúng ta có thể thấy rõ được những điểm thuận lợi khi sử dụng Bootstrap:

-Rất dễ để sử dụng: Nó đơn giản vì nó được base trên HTML, CSS và Javascript chỉ cẩn có kiến thức cơ bản về 3 cái đó là có thể sử dụng bootstrap tốt.

-Responsive: Bootstrap xây dựng sẵn reponsive css trên các thiết bị Iphones, tablets, và desktops. Tính năng này khiến cho người dùng tiết kiệm được rất nhiều thời gian trong việc tạo ra một website thân thiện với các thiết bị điện tử, thiết bị cầm tay.

-Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Tuy nhiên, với IE browser, Bootstrap chỉ hỗ trợ từ IE9 trở lên. Điều này vô cùng dễ hiểu vì IE8 không support HTML5 và CSS3.

**1.2 Angular**

-Việc phát triển Web đã có bước thay đổi đáng kể trong vài năm qua. Với phiên bản ECMAScript (ES) 2015 - chúng ta quen thuộc với cái tên ES6, với những class hay arrow function. Angular 2+ ứng dụng những tính năng mới này giúp việc code với Angular trở nên rõ ràng và dễ học hơn rất nhiều.

-Thêm vào đó, với việc ứng dụng Typescript - một ngôn ngữ - hay là một bản nâng cấp đáng giá của Javascript, Angular kết hợp với Typescript, chúng ta có một công cụ tuyệt vời giúp xử lý các vấn đề hạn chế của JS như kiểm tra kiểu dữ liệu, refactor code an toàn hơn,... từ đó cũng hỗ trợ tốt hơn cho việc Debug cũng như giúp các Dev thực sự hiểu rõ mã nguồn của họ hơn.

-Điều quan trọng của một Frameworks đối với lập trình viên đó là cấu trúc phát triển ứng dụng của nó, và Angular mang đến một kiến trúc rất rõ ràng, dựa trên ba yếu tố chính: class, các dependency được thêm vào và mô hình MVVM (model-view-view/model).

-Angular sử dụng class trong ES6 với một loạt các thuộc tính để xây dựng toàn bộ các cấu trúc chủ chốt, giả sử bạn muốn tạo một Angular component - Tạo một class và thêm vào các thuộc tính cần thiết. Hay bạn muốn tạo một Angular module - Hãy tạo một class và thêm vào đó các thuộc tính cần thiết. Về cơ bản sẽ là như vậy, Angular cung cấp một cấu trúc rõ ràng để xây dựng từng tính năng cho ứng dụng của bạn.

-Các dependency mạnh mẽ được sử dụng trong ứng dụng khi cần thiết, và khi cần tích hợp bất kì dependency nào, như HTTP hay Router, chúng ta chỉ cần thêm nó vào bên trong constructor của class.

-Mô hình MVVM cũng giúp Angular chiếm lợi thế trong xây dựng ứng dụng client-side, thường ta sẽ có 3 điều cần quan tâm chính: đó là giao diện người dùng, mã nguồn điều khiển giao diện và mô hình dữ liệu (data) cho giao diện. Angular với MVVM phân biệt hoàn toàn rõ ràng các yếu tố trên nhờ mô hình MVVM:

-Phần giao diện (view) được định nghĩa trong một template bao hàm HTML dành cho một component nhất định. Template có thể là toàn bộ Layout hoặc bất cứ mảnh ghép nào trong Layout đó.

-Model được định nghĩa như là các thuộc tính của component class. Có thể hiểu là dữ liệu, dựa vào đó để phần View sử dụng để thực thi.

-view/model là class quản lý cả view cũng như model. Là phần code sẽ xử lý việc truy xuất dữ liệu, đồng thời thực thi các tương tác của người dùng trên view.

-Với việc ứng dụng các điểm tích cực của các thành phần trên, Angular khiến việc phát triển ứng dụng trở nên dễ dàng và hiệu quả hơn.

-Đa số các ứng dụng Web không chỉ có 1 view hay một page duy nhất, mà sẽ cung cấp nhiều view khác nhau tương ứng với với các chức năng chính. Ví dụ như một trang web với các trang giới thiệu, trang nội dung, trang chi tiết, trang đăng nhập, đăng ký,... Chúng ta sẽ cần hiển thị đúng view vào đúng thời điểm. Đó là mục đích của điều hướng (routing). Và Angular cung cấp đầy đủ tính năng cho việc này, chúng ta định nghĩa các đường dẫn (route) cho mỗi page view của ứng dụng. Và chúng ta sẽ kích hoạt route dựa trên tương tác của người dùng (user). Chúng ta có thể truyền thêm dữ liệu vào các route, giúp view hiển thị nội dung một cách dynamic, có thể bảo vệ route để người dùng chỉ có thể truy cập sau khi đã đăng nhập hoặc có quyền truy cập, có thể ngăn chặn việc người dùng ngay lập tức rời một trang khi các thao tác còn dang dở cho đến khi họ thực sự xác nhận việc rời đi hoặc lưu lại tiến trình sử dụng,...Angular đồng thời cũng hỗ trợ child-route cho việc điều hướng bên trong một route. Việc điều hướng giữa các view bên trong ứng dụng Angular thực sự rất linh hoạt và mạnh mẽ.

**2. NODE JS**

-Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

-Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Dù sao thì chúng ta cũng nên biết qua một chút chút lịch sử của thứ mà chúng ta đang học một chút chứ nhỉ? =))

-Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.

-Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.

-Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

-Rõ ràng, không phải cứ hot và mới là Nodejs làm gì cũng tốt, ví dụ như một ứng dụng cần tính ổn định cao, logic phức tạp thì các ngôn ngữ PHP hay Ruby… vẫn là sự lựa chọn tốt hơn. Còn dưới đây là những ứng dụng có thể và nên viết bằng Nodejs:

+Websocket server: Các máy chủ web socket như là Online Chat, Game Server…

+Fast File Upload Client: là các chương trình upload file tốc độ cao.

+Ad Server: Các máy chủ quảng cáo.

+Cloud Services: Các dịch vụ đám mây.

+RESTful API: đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.

+Any Real-time Data Application: bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.

**3. MONGODB**

-MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

-MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

-Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng

-So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row , MongoDB sẽ dùng các document thay cho row trong RDBMS.

-Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

-Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

***Một số câu lệnh cơ bản trên MongoDb***

|  |  |  |
| --- | --- | --- |
| CSDL | MySQL | MongoDb |
| Tạo csdl | CREATE DATABASE test; | use test; |
| Tạo bảng | CREATE TABLE students (ten\_cot - kieu\_du\_lieu); | db.createCollection('students'); |
| Tạo bản ghi | INSERT INTO studetns ('name', 'gender') VALUES('thanh', 'male'); | db.students.insert({ name:'thanh', gender: 'male'}); |
| Cập nhật | UPDATE students SET name = 'thanh update' WHERE id = 1; | db.students.update({ \_id: 1 },{$set:{ name: 'thanh update' }}); |
| Xóa bản ghi | DELETE FROM students Where id = 1; | db.students.remove({ \_id: 1}); |
| Tìm kiếm all | SELECT \* FROM students; | db.students.find({}); |
| Tìm kiếm | SELECT \* FROM students WHERE name = 'thanh'; | db.students.find({ name: 'thanh' }); |