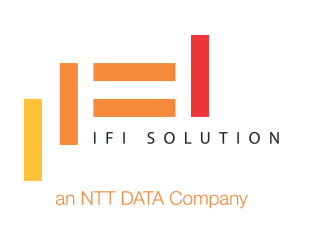
1. ĐẠI HỌC QUỐC GIA HÀ NỘI
2. TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



1. TÊN ĐỀ TÀI  
   Hệ thống thi trắc nghiệm kiểm tra đầu vào cho của ifi solution
2. Cán bộ hướng dẫn: Đinh Văn Hoàn, Phạm Văn Tùng
3. Người thực hiện: Đặng Sơn Tùng

HÀ NỘI – 2021

Mục lục

[Mục lục 1](#_Toc79078113)

[Chương 1. Giới thiệu 2](#_Toc79078114)

[1.1. Đặt vấn đề 2](#_Toc79078115)

[1.2. Giới thiệu bài toán 2](#_Toc79078116)

[1.3. So sánh với giải pháp đã có 2](#_Toc79078117)

[Chương 2. Kiến trúc và công nghệ sử dụng 3](#_Toc79078118)

[2.1. Công nghệ 3](#_Toc79078119)

[2.1.1. Frontend 3](#_Toc79078120)

[2.1.2. Backend 4](#_Toc79078121)

[2.1.3. Cơ sở dữ liệu 4](#_Toc79078122)

[2.2. Kiến trúc 5](#_Toc79078123)

[Chương 3. Cài đặt và khởi động hệ thống 6](#_Toc79078124)

[3.1. Cài đặt source code về local 6](#_Toc79078125)

[3.2. Cơ sở dữ liệu 6](#_Toc79078126)

[3.3. Backend 7](#_Toc79078127)

[3.4. Frontend 7](#_Toc79078128)

[Chương 4. Phân tích và thiết kế hệ thống 8](#_Toc79078129)

[4.1. Ca sử dụng 8](#_Toc79078130)

[4.2. Đặc tả yêu cầu 8](#_Toc79078131)

[4.3. Phân tích đặc tả yêu cầu 10](#_Toc79078132)

[4.4. Thiết kế cơ sở dữ liệu 10](#_Toc79078133)

[Chương 5. Kết quả đạt được 11](#_Toc79078134)

[Kết luận 23](#_Toc79078135)

[Tài liệu tham khảo 24](#_Toc79078136)

# Giới thiệu

## Đặt vấn đề

Vào mỗi đợt tuyển dụng, công ty sẽ chọn ra các ứng viên phù hợp để ứng tuyển vào các vị trí Fresher như Java, DevOpt, … để tham gia vào chương trình đào tạo 3 tháng. Trong 3 tháng đó, các cán bộ hướng dẫn sẽ dạy và cho Fresher tìm hiểu những kiến thức về lập trình từ Backend đến Frontend. Sau mỗi lần học xong 1 khung chương trình nhất định thì sẽ có bài kiểm tra cho mỗi Fresher bằng hình thức hỏi đáp với Cán bộ hướng dẫn hoặc lúc tuyển dụng thì người ứng tuyển sẽ có 1 bài kiểm tra bằng giấy để đánh giá và chấm điểm đầu vào để xem ứng viên đó có đủ điều kiện để thành Fresher của công ty không

Phương pháp hiện tại tồn tại nhiều nhược điểm:

* Tình hình dịch covid-19 đang diễn biến hết sức phức tạp, nhà nước chính quyền cũng ra chỉ thị đối với mọi người là hạn chế ra khỏi nhà. Với việc đến tận công ty để kiểm tra qua giấy hoặc hỏi đáp thì rất là khó trong tình hình dịch bệnh.
* Khi ứng viên đã làm xong bài kiểm tra thì lại phải mất thâm nhân lực bỏ thời gian đẻ chấm điểm bài làm hoặc sau khi hỏi đáp xong thì CBHD (Cán bộ hướng dẫn) có thể quên kết quả đánh giá của Fresher trước nếu không ghi chép lại vì quá trình hỏi đáp diễn ra liên tục người này thay người kia.
* Hệ thống thi trắc nghiệm đánh giá đầu vào của IFI Solution ra đời để giải quyết những vấn đề trên.

## Giới thiệu bài toán

Em sẽ nghiên cứu để xây dựng lên 1 website có thể dùng để ra các bài kiểm tra kiến thức lập trình hoặc tiếng Anh để có thể đánh giá tầm hiểu biết của Fresher.

## So sánh với giải pháp đã có

* Hệ thống đơn giản rất nhiều công đoạn so với quy trình kiểm tra đánh giá đầu vào cho ứng viên/Fresher
* Tiết kiệm tài nguyên (giấy, tiền in phiếu, ...)
* Tiết kiệm công sức, thời gian của cán bộ hướng dẫn vì cán bộ hướng dẫn có thể bận với dự án của công ty không thể dành quá nhiều thời gian để đánh giá Fresher được.

# Kiến trúc và công nghệ sử dụng

## Công nghệ

### Frontend

* **HTML/CSS/Javascript**:
* **HTML (Hyper Text Markup Language):** cho phép người dùng tạo và cấu trúc hóa các thành phần trên một trang web. Các phần tử trong HTML là các khối của trang web HTML, được đại diện bằng những thẻ đánh dấu (tag) có chứa nội dung cụ thể của từng thẻ.
* **CSS (Cascading Style Sheets):** là một ngôn ngữ được dùng để tìm kiếm và định dạng lại phần tử tạo ra bằng các ngôn ngữ đánh dấu như HTML. Các phần tử HTML sẽ được định nghĩa kiểu và cách hiển thị. Hỗ trợ lập trình viên phân tách giữa nội dung và cách trình bày trang.
* **JS (Javascript):** là ngôn ngữ kịch bản dùng để thực hiện được các thuật toán logic, do đó hoàn toàn có thể thực hiện xác nhận, tính toán, thao tác với dữ liệu.
* **Bootstrap 4**: là framework của HTML, CSS, và JavaScript (bao gồm các templates của HTML, CSS và Javascript tạo ra những cái cơ bản có sẵn) Bootstrapcho phép người dùng sử dụng để thiết kế website dễ dàng với 1 chuẩn nhất định, tạo các trang website thân thiện và hỗ trợ cho việc thiết kế reponsive cho website thuận tiện, dễ dàng và nhanh chóng hơn.
* **jQuery**: là thư viện được viết từ JavaScript. jQuery giúp người dùng xây dựng các chức năng một cách dễ dàng, nhanh và giàu tính năng hơn bằng Javascript . jQuery cung cấp các API giúp người dùng như: duyệt tài liệu HTML, hoạt ảnh, xử lý sự kiện và thao tác AJAX đơn giản hơn.
* **Angular (version 12)**: là một framework cho frontend và là bản tiếp theo của AngularJS. Angular là mã nguồn mở giúp chúng ta xây dựng một Single Page Applications (SPAs). Bao gồm các tính năng nổi bật như:
* Cơ chế Two-Way Data Binding: bất cứ thay đổi nào trên view đều được tự động cập nhật vào component class và ngược lại.
* Hỗ trợ cơ chế Routing mạnh mẽ: Angular có cơ chế routing tải trang một cách bất đồng bộ trên cùng một trang cho phép chúng ta tạo SPA.
* Mở rộng HTML: Angular cho phép chúng ta sử dụng cấu trúc lập trình giống như điều kiện if, vòng lặp for...để render các control.
* Thiết kế module hoá: Angular tiếp cận theo hướng thiết kế module hoá. Bạn phải tạo các Angular Module để tổ chức tốt hơn và quản lý source code.
* Hỗ trợ làm việc với hệ thống Backend: Angular được xây dựng hỗ trợ làm việc với backend server và thực thi bất cứ logic nào và nhận dữ liệu về.
* **Một số Module tích hợp sử dụng thêm từ npm:** ngx-countdown, ng2-search-filter, ng-select, angular-file-uploader, ng-bootstrap

### Backend

* **Java Spring Framework**: là một Framework để phát triển ứng dụng Java.
* Lý do sử dụng:
* Nhẹ: Spring framework hỗ trợ POJO. Vì vậy sẽ không bao giờ cần đến EJB (Enterprise Java Beans - thường cần sử dụng trong các Framework khác). Trên thực tế, phiên bản Spring basic chỉ nặng khoảng 2 MB Có thể áp dụng cho bất kỳ ứng dụng Java nào
* Spring framework rất thân thiện với các lập trình viên. Nó có thể kiểm thử dễ dàng, nhẹ, thích ứng và dễ cấu hình
* Tính Module mạnh mẽ: Spring framework là một framework thích ứng. Nó cung cấp nhiều mô-đun mạnh mẽ
* Các mô-đun sử dụng trong dự án:

1. Spring Boot để khởi tạo và chạy dự án
2. Spring Core: những gì cốt lõi hình thành framework, có thể kể đến: IoC, Application Context, annotation, …
3. Spring MVC: hỗ trợ kiến trúc MVC (Model-View-Controller) và các component để phát triển các ứng dụng.
4. Spring Data JPA: triển khai tư tưởng ORM (Object Relational Mapping): tư tưởng ánh xạ CSDL sang các đối tượng.

### Cơ sở dữ liệu

* **Mysql**:
* là một hệ thống cơ sở dữ liệu với mã nguồn mở **(RDBMS - Relational Database Management System)** hoạt động theo mô hình **client-server.**
* là cơ sở dữ liệu có quan hệ, làm nổi bật tính hướng đối tượng của đối tượng trong hệ thống

## Kiến trúc

**MVC** là từ viết tắt bởi 3 từ **Model - View - Controller**. Đây là một mô hình trong thiết kế sử dụng hệ thống. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ sẽ tương ứng với một hoạt động trong mô hình.



Hình 1: Mô hình MVC

Mô hình MVC và các thành phần bên trong của MVC.

* **Model(M):** Là bộ phận có lưu trữ tất cả dữ liệu của hệ thống. **Model** sử dụng như một cầu nối giữa 2 thành phần bên dưới là Controller và View. Model được thể hiện như một cơ sở dữ liệu hoặc có khi chỉ là một [file XML](https://monamedia.co/file-xml-la-gi/) bình thường.
* **View (V):** Đây là phần giao diện (theme) dành cho người sử dụng. View là nơi giúp tương tác với người dùng như lấy được thông tin dữ liệu thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website.
* **Controller (C): là** bộ phận trung gian với nhiệm vụ xử lý các yêu cầu người dùng gửi đến thông qua View. Qua đó, **Controller sẽ** đưa ra dữ liệu phù hợp với người dùng. Controller còn có chức năng kết nối với model.

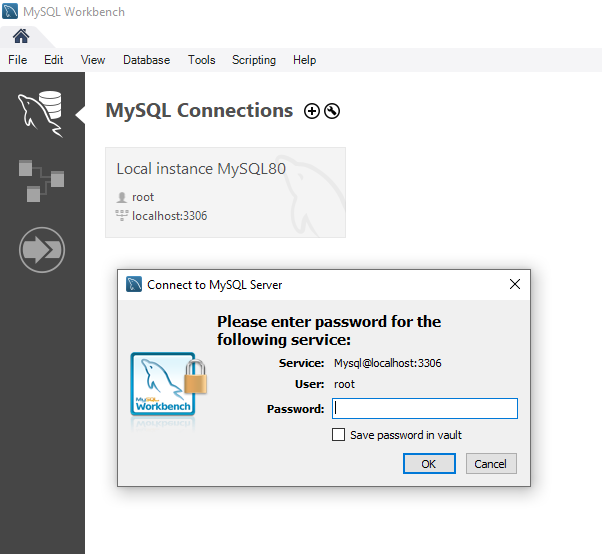
# Cài đặt và khởi động hệ thống

## Cài đặt source code về local

* Download và cài đặt Git
* Clone code về local: git clone https://github.com/TungSoDa/IFI\_Work.git

## Cơ sở dữ liệu

* Download và cài đặt MySQL: <https://dev.mysql.com/downloads/installer/>
* Sau khi cài đặt xong mở MySQL Workbench, đăng nhập bằng mật khẩu khi cài đặt



Hình 2: Truy cập vào cơ sở dữ liệu trên local

* Tạo 1 schema mới, để tên là **ifi\_fresher\_test**
* Chọn *Open a SQL script file in a new query tab* rồi tìm đến thư mục **database** trong folder IFI\_Fresher\_Test sau khi clone code về
* Chọn và execute từng file trong schema **ifi\_fresher\_test,** ta sẽ có 7 bảng là **account, answer, contestant, contributor, exam, exam\_result, question**

## Backend

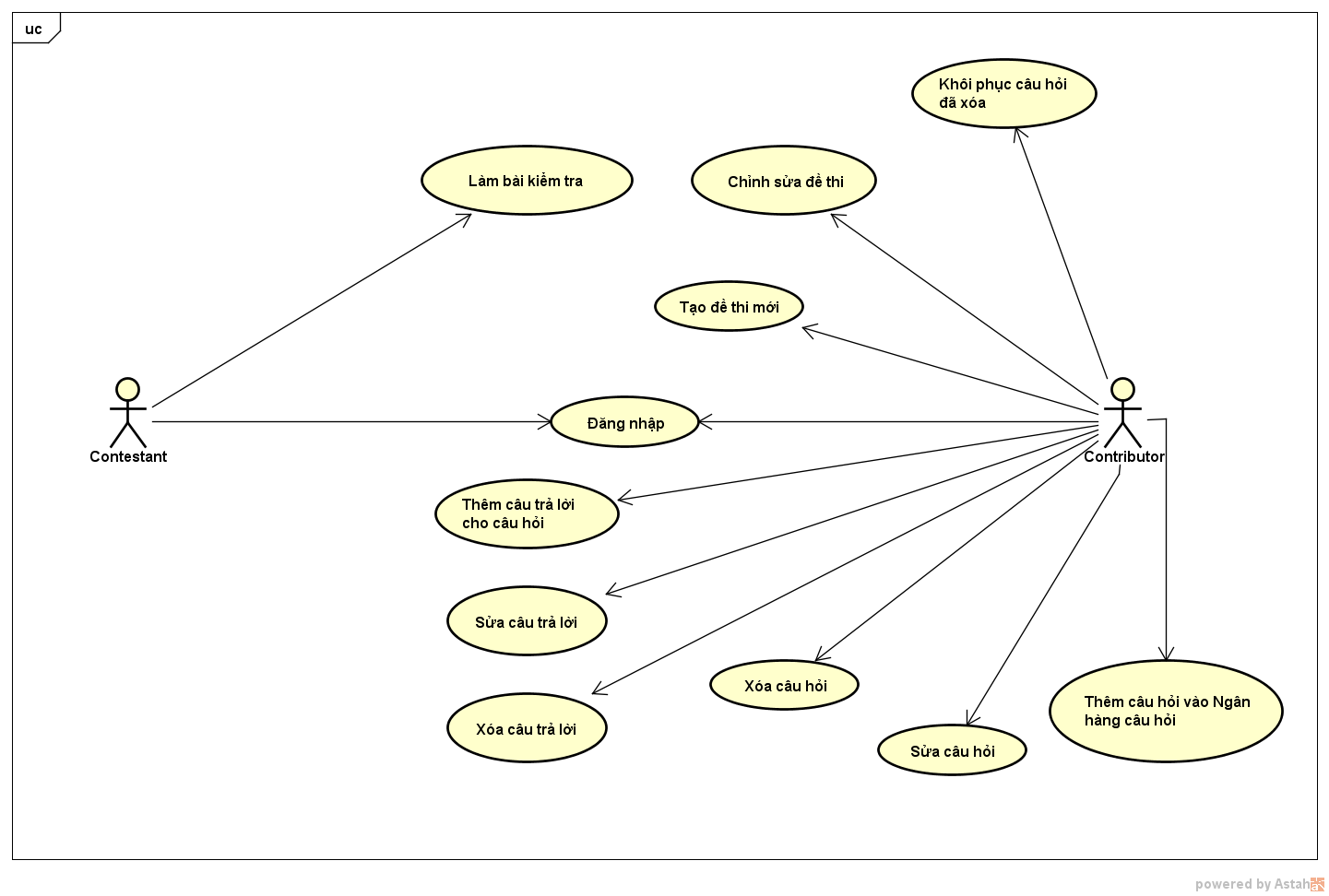
* Download và cài đặt IDE Spring Tool Suite for Eclipse: <https://spring.io/tools>
* Mở và chạy project: Chọn File ⇨ Import ⇨ Maven ⇨ Existing Maven Projects ⇨ Tìm đường dẫn đến folder IFI\_Fresher\_Test khi clone từ git về
* Chọn file *pom.xml* ⇨ Run as ⇨ Maven clean ⇨ Maven install
* Vào thư mục src/main/java/com.ifi.fresher\_test.ifi\_fresher\_test và chọn file IfiFresherTestApplication.java ⇨ Run As ⇨ Spring Boot App
* Hoặc với IntelliJ IDEA Ultimate: <https://www.jetbrains.com/idea/download/>
* Mở và chạy project: Chọn Open ⇨ Tìm đường dẫn đến folder IFI\_Fresher\_Test khi clone từ git về ⇨ Run

## Frontend

* Download cài cài đặt VS Code: <https://code.visualstudio.com/download>
* Mở và chạy project: Chọn File ⇨ Open Folder ⇨ Tìm đường dẫn đến folder IFI\_Fresher\_Test khi clone từ git về
* Mở Terminal chạy lệnh: cd src\main\resources\ifi-fresher-test ⇨ ng serve
* Mở browser vào đường dẫn: localhost:4242

# Phân tích và thiết kế hệ thống

## Ca sử dụng



Hình 3: Biểu đồ ca sử dụng tổng quát

## Đặc tả yêu cầu

* Yêu cầu chức năng

Bảng 1: Yêu câu chức năng

|  |  |  |
| --- | --- | --- |
| STT | Yêu cầu | Mô tả |
| 1 | Đăng nhập | Người dùng đăng nhập vào hệ thống với tài khoản và mật khẩu được cung cấp |
| 2 | Làm bài kiểm tra | Fresher sau khi đăng nhập vào hệ thống thì chọn đề thi theo chủ đề được cán bộ yêu cầu để làm bài kiểm tra |
| 3 | Tạo đề thi mới | CBHD tạo một đề thi mới bằng cách chọn chủ đề của đề thi và chọn câu hỏi từ ngân hàng câu hỏi hoặc để hệ thống tạo câu hỏi tự động |
| 5 | Chỉnh sửa đề thi | CBHD chọn đề thi muốn sửa trong Kho đề thi, có thể đổi tên, các câu hỏi cùng chủ đề của đề thi đó |
| 6 | Thêm câu hỏi vào ngân hàng câu hỏi | CBHD nhập nội dung, chọn chủ đề và them hình ảnh nếu có cho câu hỏi và thêm vào Ngân hàng câu hỏi |
| 7 | Sửa câu hỏi | CBHD chọn câu hỏi muốn sửa và nhập nội dung, chọn chủ đề và tải hình ảnh muốn thay đổi cho câu hỏi đó và Lưu lại thay đổi |
| 8 | Xóa câu hỏi | CBDH chọn câu hỏi muốn xóa và xác nhận là Xóa và câu hỏi đó sẽ được chuyển đến Câu hỏi đã xóa |
| 9 | Thêm câu trả lời cho câu hỏi | CBHD chọn câu hỏi muốn thêm và nhập nội dung và chọn câu trả lời đó là đáp án đúng hay không |
| 10 | Sửa câu trả lời | CBHD chọn câu hỏi muốn sửa và nhập nội dung và chọn câu trả lời đó là đáp án đúng hay không |
| 11 | Xóa câu trả lời | CBHD chọn câu hỏi muốn xóa và xác nhận là Xóa câu hỏi đó |
| 12 | Khôi phục câu hỏi đã xóa | CBHD vào Câu hỏi đã xóa, chọn và xác nhận khôi phục đã xóa |

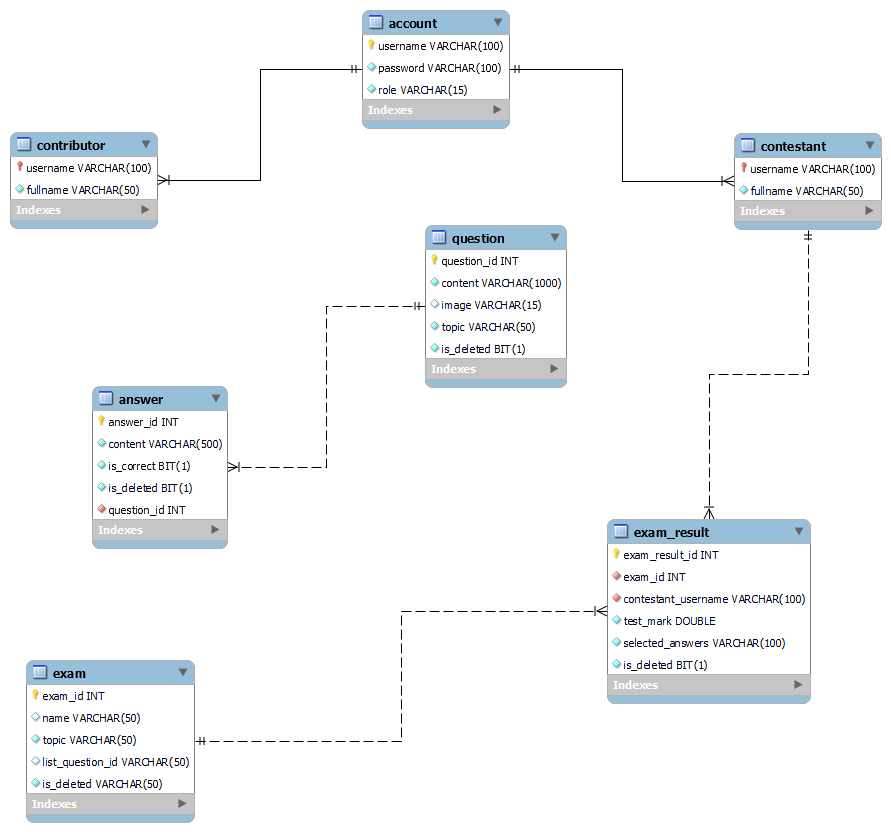
* Yêu cầu phi chức năng

Bảng 2: Yêu câu phi chức năng

|  |  |
| --- | --- |
| **STT** | **Yêu cầu** |
| **1** | * Giao diện thân thiện * Dễ sử dụng * Tiệm cận với các hệ thống đã có sẵn của trường |
| **2** | Thời gian phản hồi ngắn. |
| **3** | Website hoạt động ổn định, tránh tình trạng tắc nghẽn khi có nhiều người dung cùng 1 lúc. |
| **4** | Thông tin tài khoản được bảo mật. |
| **5** | Dữ liệu được cập nhật theo thời gian thực. |

## Phân tích đặc tả yêu cầu

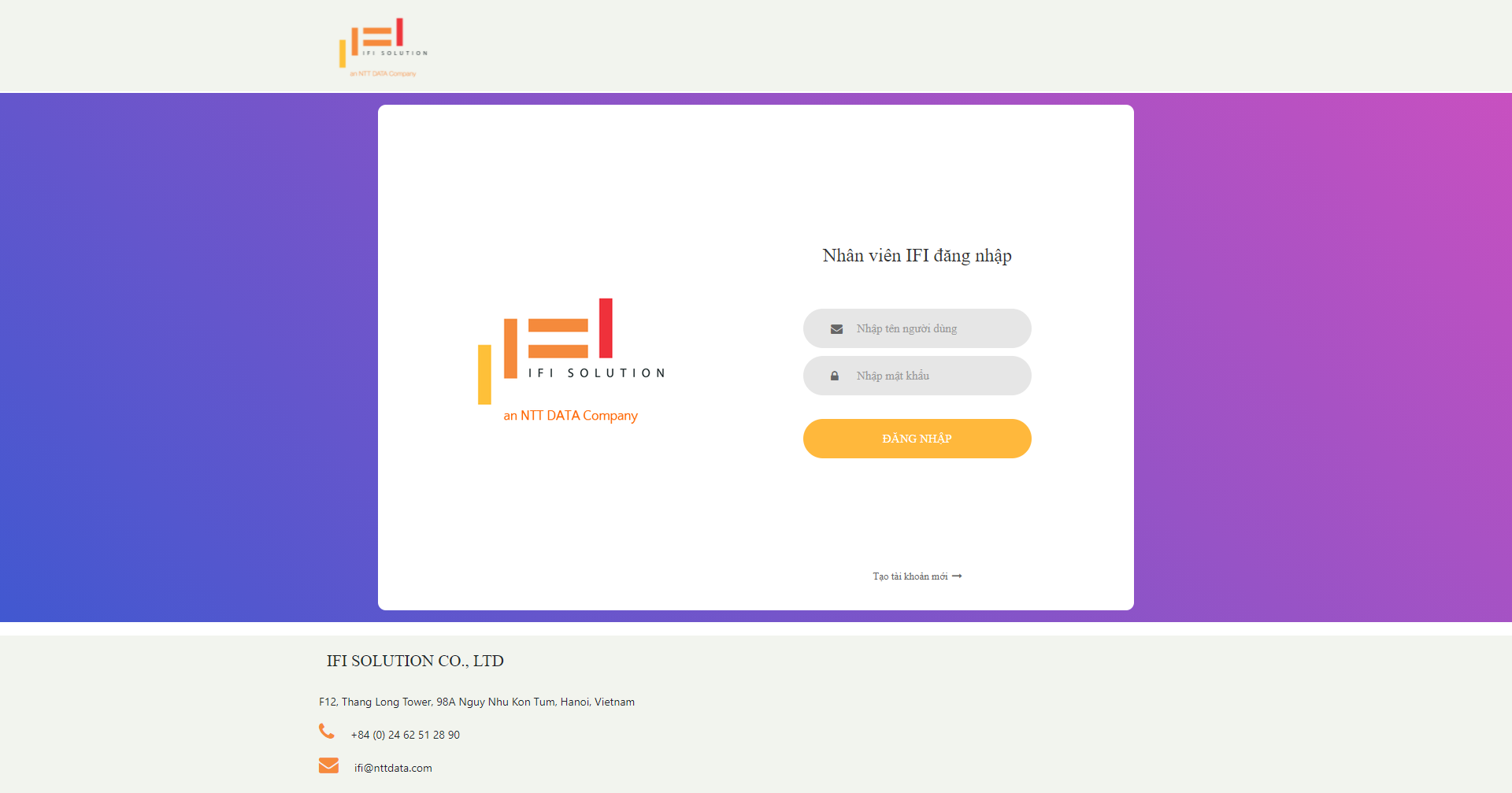
## Thiết kế cơ sở dữ liệu



Hình 4: Lược đồ Quan hệ - Thực thể

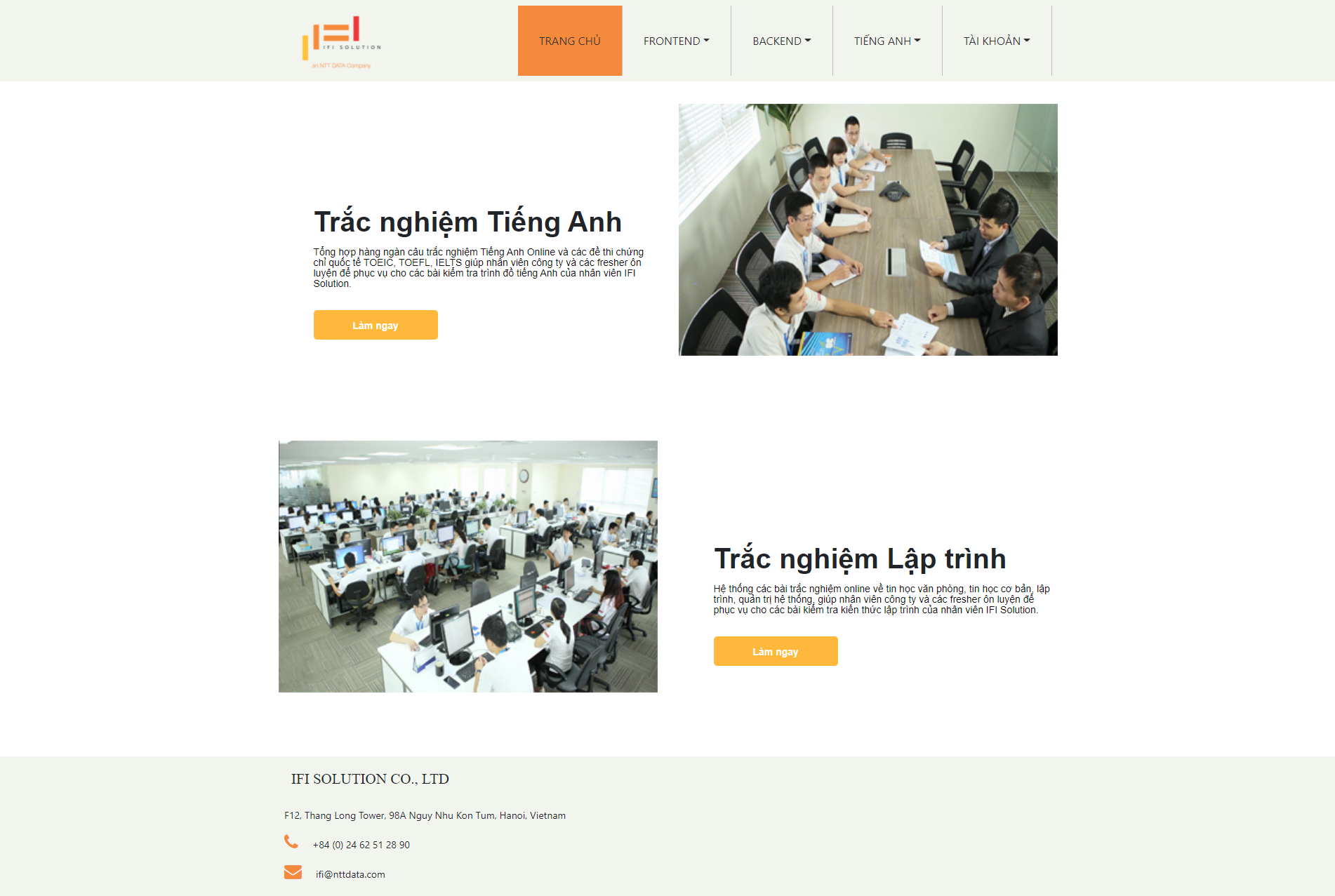
# Kết quả đạt được

Dưới đây là các màn hình của hệ thống thi trắc nghiệm đầu vào trong quá trình chạy



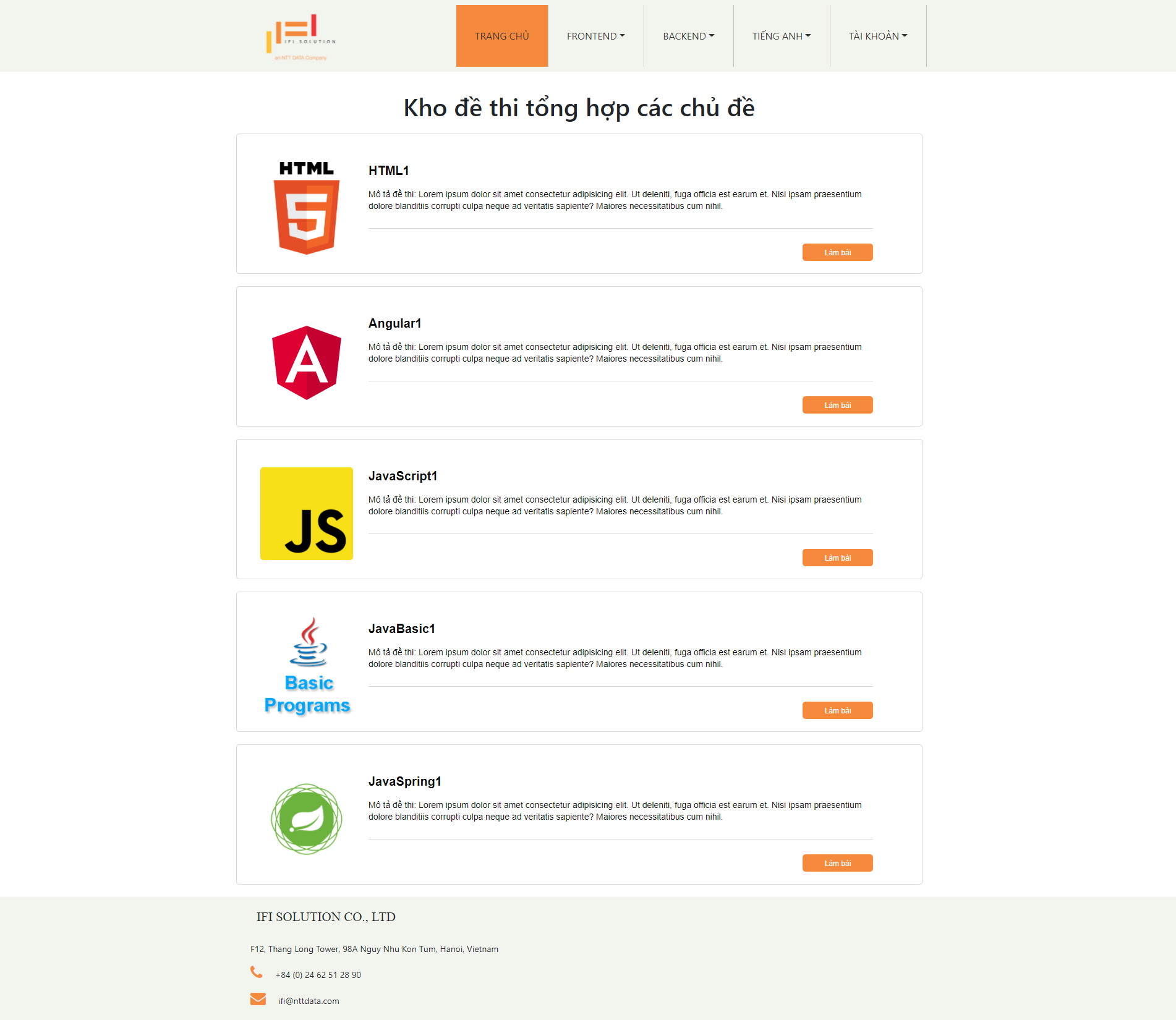
Hình 5: Giao diện trang Đăng nhập

* Với người dùng hệ thống (fresher) đăng nhập bằng tài khoản: tungds/TungDS@1234
* Với quản trị viên (cán bộ hướng dẫn) đăng nhập bằng tài khoản: tungpv/TungPV@1234



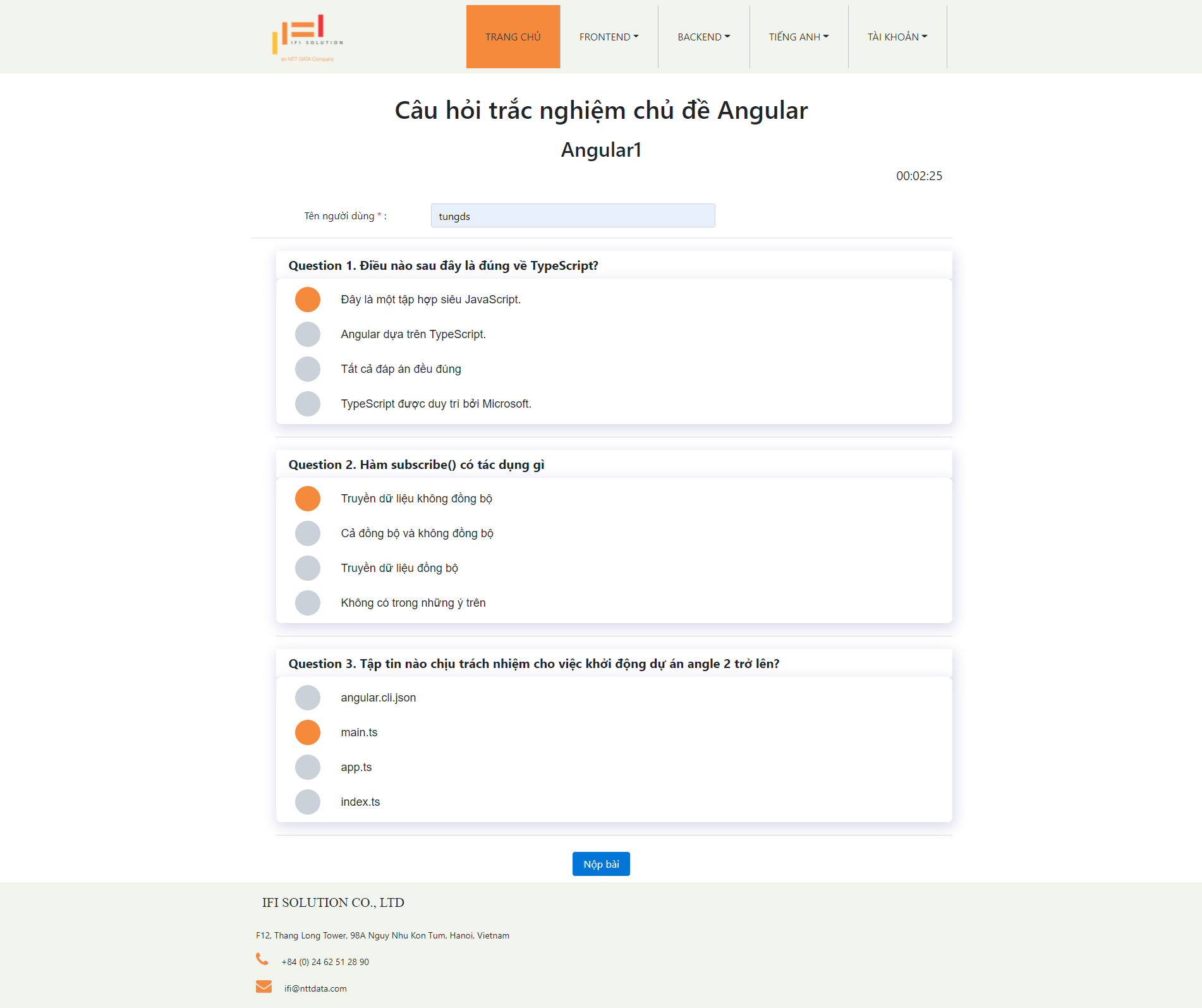
Hình 6: Giao diện trang chủ của người dùng hệ thống (contestant)

* Chọn Làm ngay ở phần Trắc nghiệm Lập trình



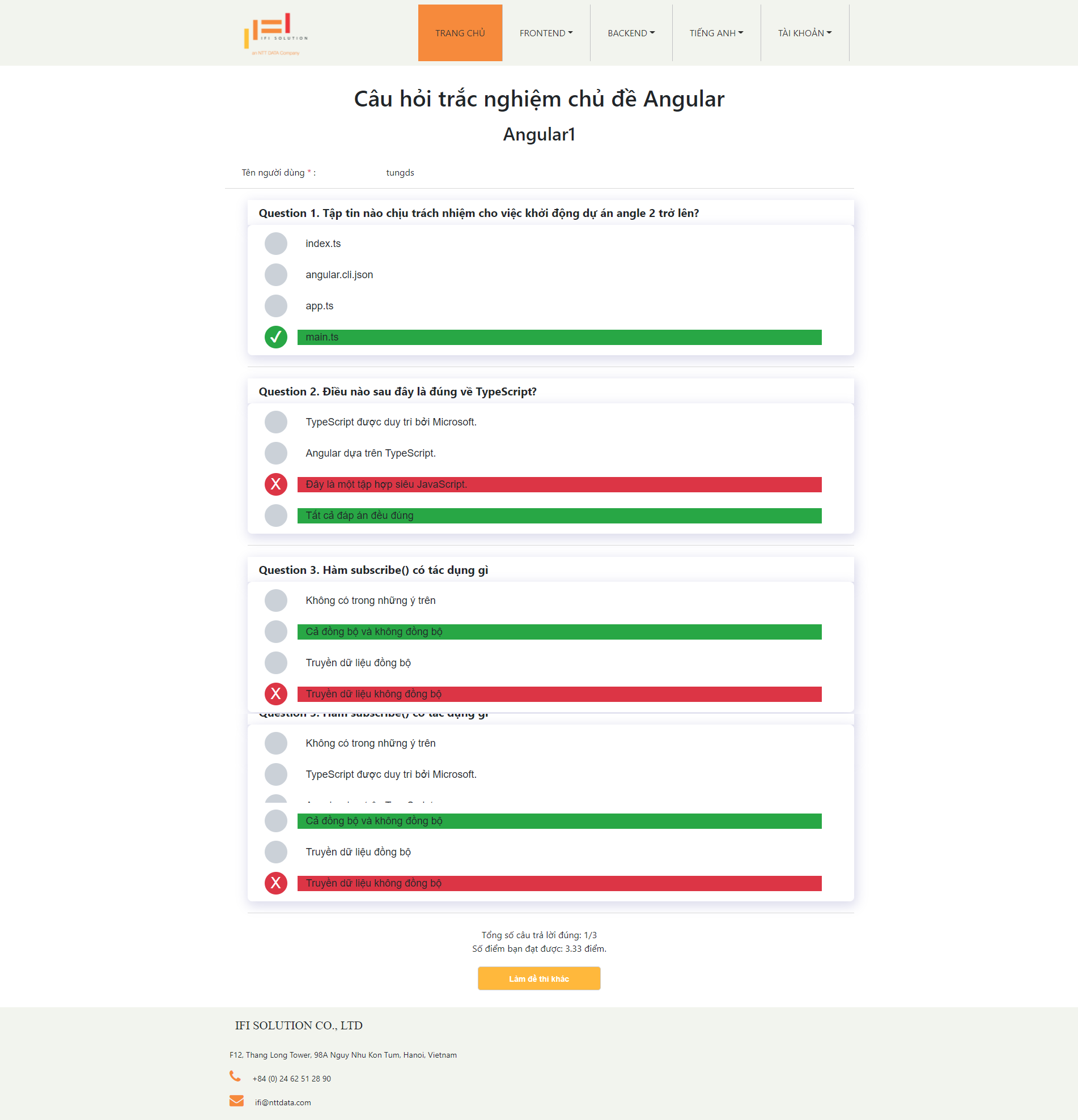
Hình 7: Giao diện Kho đề thi tất cả các chủ đề

* Chọn đề thi muốn làm và click Làm bài



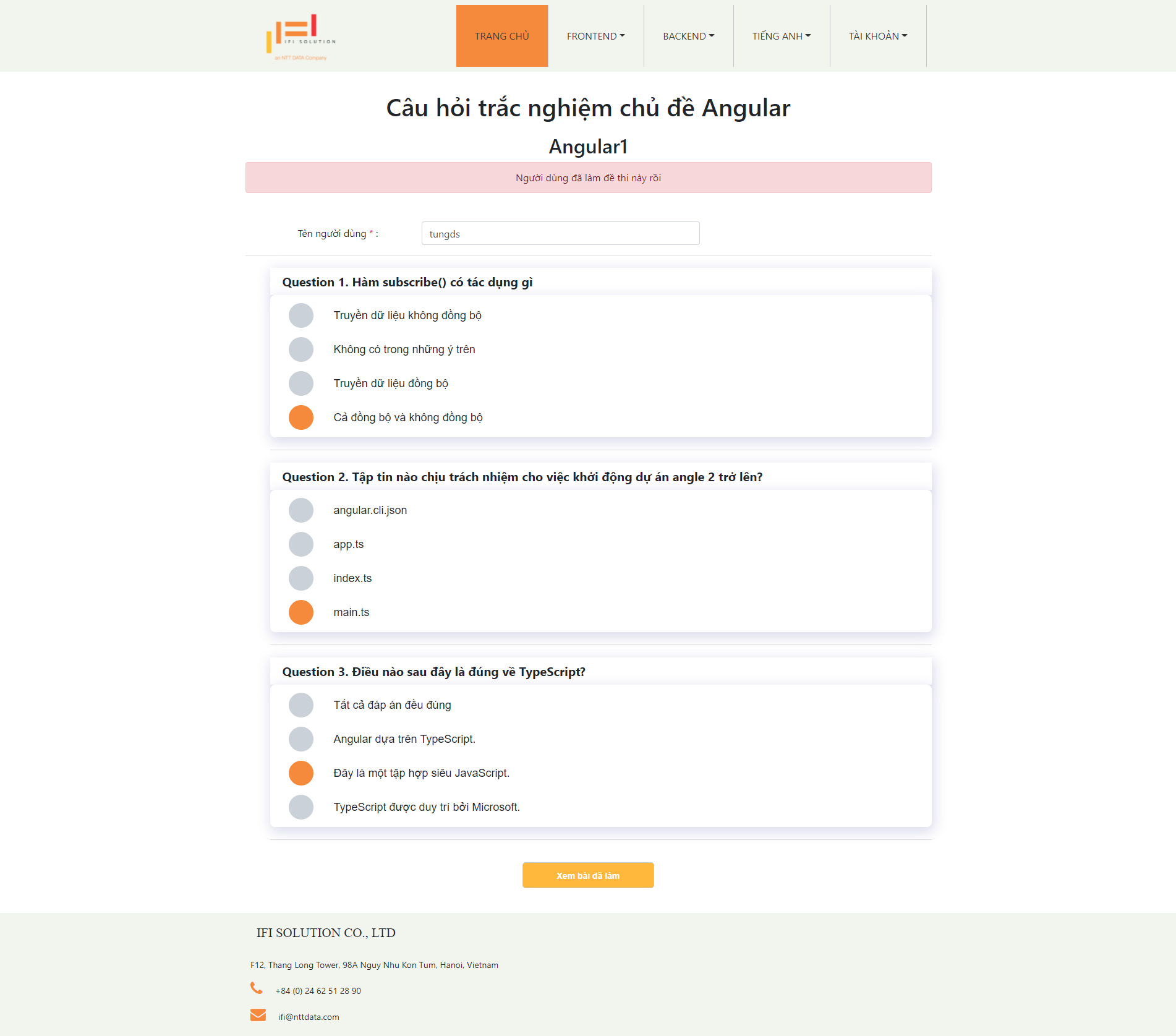
Hình 8: Giao diện đề thi trắc nghiệm

* Người làm điền tên người dùng mà đã dùng để đăng nhập vào hệ thống và chọn đáp án cho từng câu hỏi. Sau khi làm xong nhấn Nộp bài, thời gian hết thì bài kiểm tra sẽ được tự động nộp bài và chấm điểm.
* Nếu người làm không điền tên người dùng hay chưa chọn đáp án thì hệ thống sẽ có thông báo.



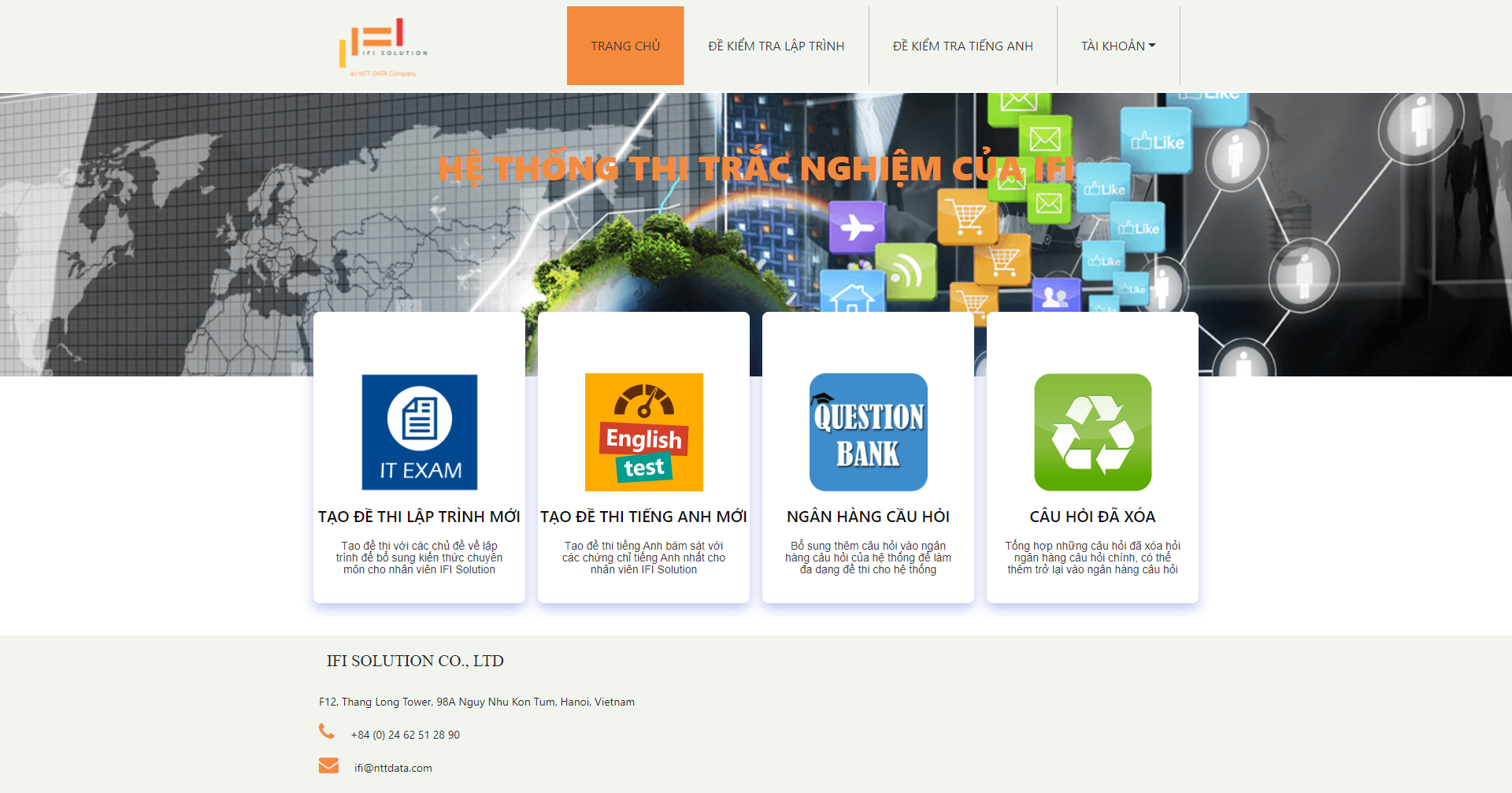
Hình 9: Giao diện kết quả bài kiểm tra

* Sau khi nộp bài hệ thống sẽ chấm điểm và trả về kết quả bài kiểm tra đã làm. Với câu hỏi chọn đúng sẽ được bôi xanh còn câu hỏi chọn sai sẽ bôi đỏ và bôi xanh đáp án đúng của câu hỏi đó

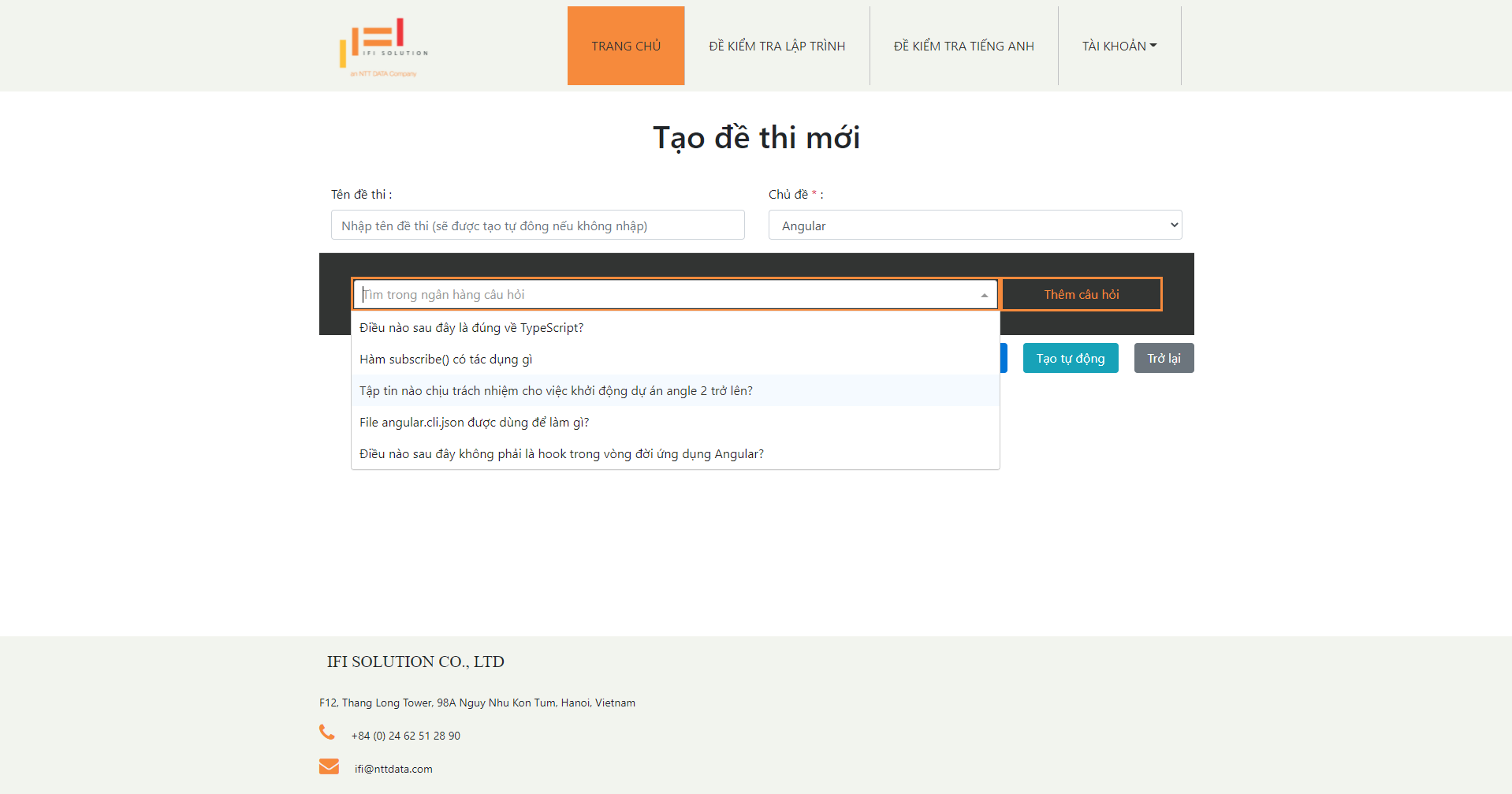


Hình 10: Giao diện báo lỗi khi người dùng đã làm đề thi

* Nếu người dùng đã làm đề thi thì khi nộp bài sẽ có thông báo



Hình 11: Giao diện trang chủ của quản trị viên (contributor)



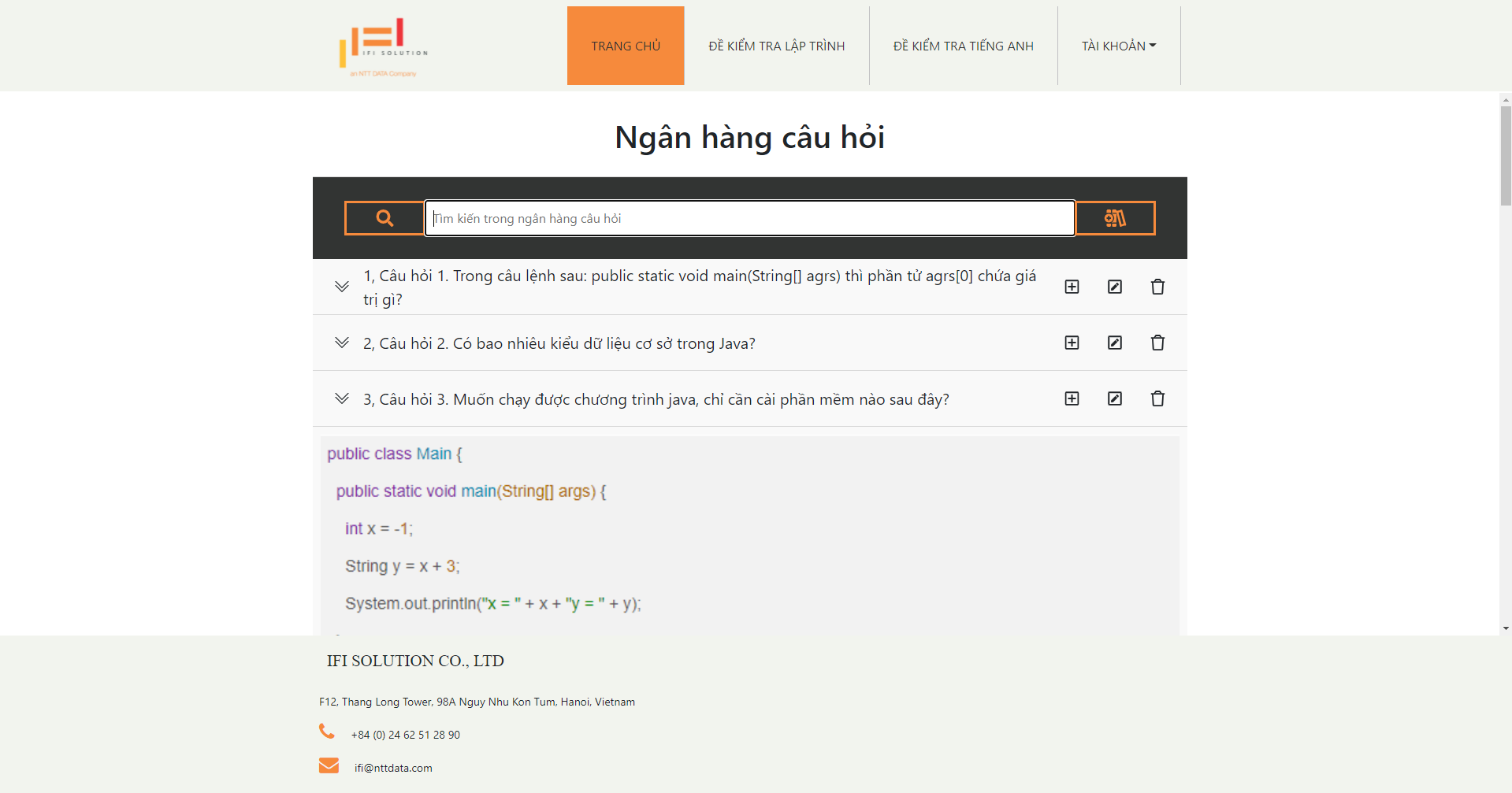
Hình 12: Giao diện Tạo đề thi mới

* CBHD nhập tên cho đề thi, nếu không nhập tên thì tên sẽ được tạo tự đông theo chỉ số (VD: Topic1, Topic2, Topic3, …)
* CBHD chọn chủ đề cho đề thi, danh sách câu hỏi ở phần Tìm kiếm sẽ được thay đổi theo đúng chủ đề được chọn (chỉ load ra những câu hỏi của chủ đề đã chọn)
* Chọn 1 câu hỏi trong danh sách và nhấn Thêm câu hỏi
* Nếu thêm câu hỏi đã thêm thì sẽ báo lỗi



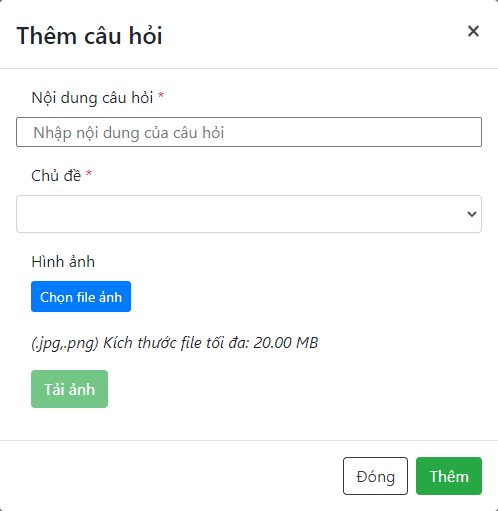
Hình 13: Nút chức năng của giao diện tạo đề thi

* Sau khi chọn xong danh sách câu hỏi phù hợp thì nhấn Tạo để tạo đề thi mới
* Hiện tại đề thi 1 chủ đề là 3 câu và đề thi tổng hợp là 18 câu, nếu chọn không đúng số lượng câu hỏi thì sẽ có thông báo
* Nếu đã tồn tại đề thi cầu chủ đề và có list câu hỏi tương tự (theo bất kì thứ tự nào) thì sẽ có thông báo
* Nếu không muốn chọn câu hỏi thì có thể nhấn Tạo tự động, hệ thống sẽ chọn random câu hỏi trong danh sách câu hỏi tương ứng và tạo thành 1 đề thi mới



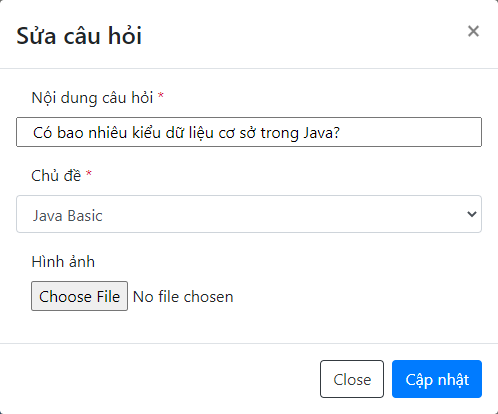
Hình 14: Giao diện Ngân hàng câu hỏi

* Chọn icon ở cạnh ô tìm kiếm để thêm câu hỏi mới
* Di chuột vào câu hỏi để xem các câu trả lời
* Chọn icon đầu tiên sau nội dung câu hỏi để thêm câu trả lời, thứ 2 là để Sửa câu hỏi và thứ 3 là để xóa câu hỏi

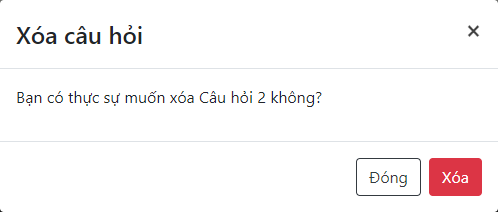


Hình 15: Dialog thêm câu hỏi

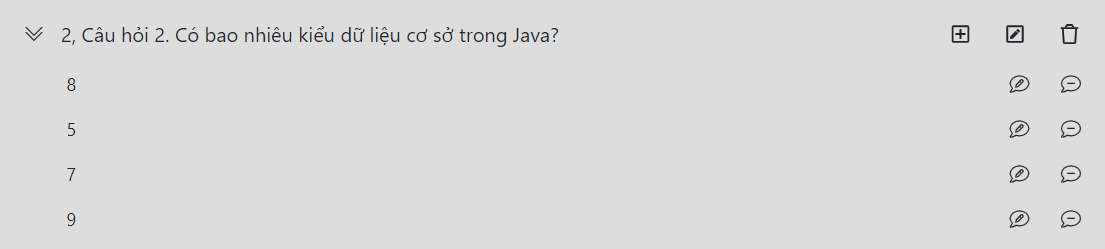
* CBHD nhập nội dung câu hỏi, chọn chủ đề cho câu hỏi
* Hiện tại chức năng chọn hình ảnh chưa xong vì chưa lưu được xuống folder trong project
* Nếu nhập nội dung câu hỏi đã có trong Ngân hàng câu hỏi thì sẽ báo lỗi



Hình 16: Dialog sửa câu hỏi

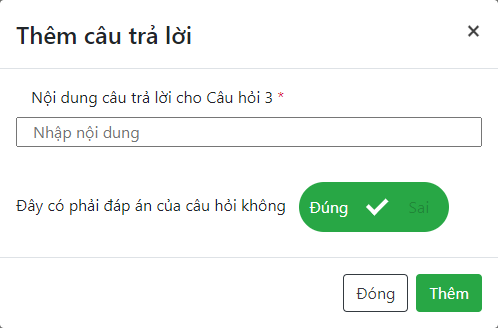


Hình 17: Dialog xóa câu hỏi



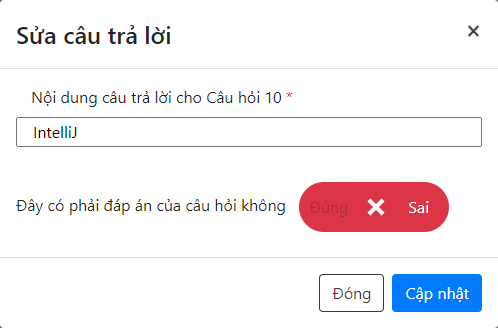
Hình 18: Các câu trả lời trong câu hỏi

* Nhấn icon đầu tiên sau nội dung câu trả lời để sửa câu trả lời và thứ 2 để xóa câu trả lời

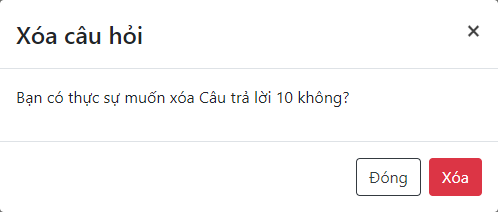


Hình 19: Dialog thêm câu trả lời

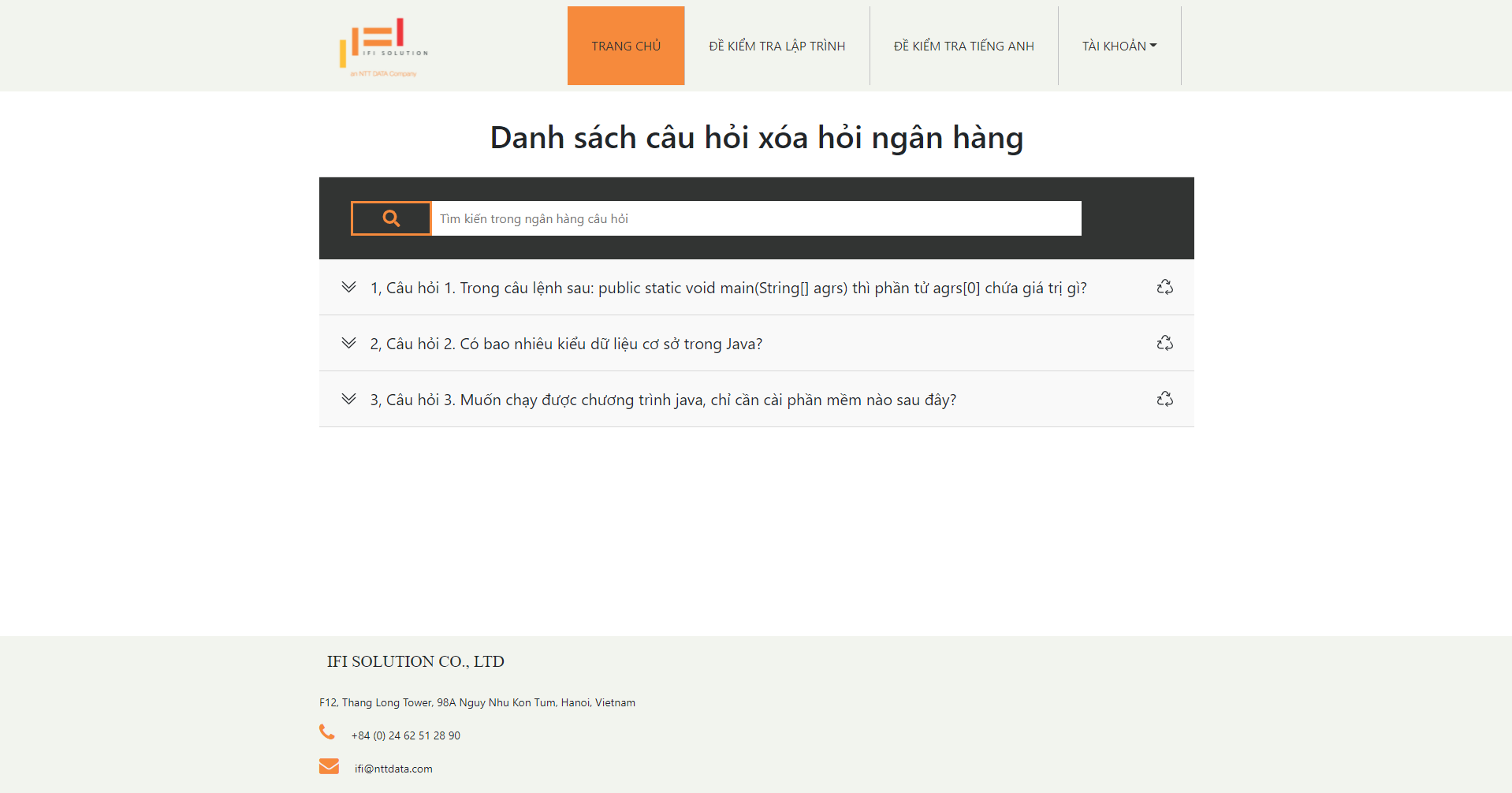
* CBHD nhập nội dung câu trả lời và chọn câu trả lời đó là đúng hay sai
* Nếu nội dung trùng với nội dung câu trả lời khác trong cùng câu hỏi thì sẽ có thông báo.
* Nếu đã có đáp án đúng trong câu hỏi mà đáp án thêm mới chọn đúng thì sẽ có thông báo



Hình 20: Dialog sửa câu trả lời

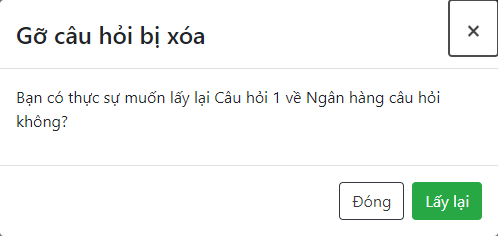


Hình 21: Dialog xóa câu trả lời



Hình 22: Giao diện Câu hỏi đã xóa

* Nhấn vào icon sau nội dung câu hỏi để khôi phục câu hỏi bị xóa



Hình 23: Dialog khôi phục câu hỏi bị xóa

Kết luận

Qua quá trình làm project, em cũng đã học hỏi thêm được nhiều kiến thức mới mẻ. Học được them về nhiều công nghệ mới, các kiến thức mới về lập trình, tuy làm 1 mình nhưng cũng học hỏi từ các bạn Fresher khác các làm việc nhóm, cách tạo nên cấu trúc hợp lý của 1 project

Về mặt kiến thức, củng cố, nâng cao kiến thức về mảng web: HTML5, CSS3, Javascript, Bootstrap 4, Angular, Java Spring.

Hệ thống đã đáp ứng được các yêu cầu cơ bản, tuy nhiên hệ thống vẫn còn nhiều nhược điểm.

Hướng phát triển của dự án:

* Mở rộng thêm các thể loại câu hỏi: điền đáp án, chọn nhiều đáp án trong 1 câu hỏi.
* Thiết lập các tài khoản “ảo” dùng cho ứng viên khi tuyển dụng. Nếu được Cán bộ phê duyệt là pass phỏng vấn thì tài khoản sẽ chuyển thành tài khoản Fresher và dùng để đánh giá đầu vào tiếp theo.
* Bổ sung thêm Spring Security để phân quyền cho người dùng.