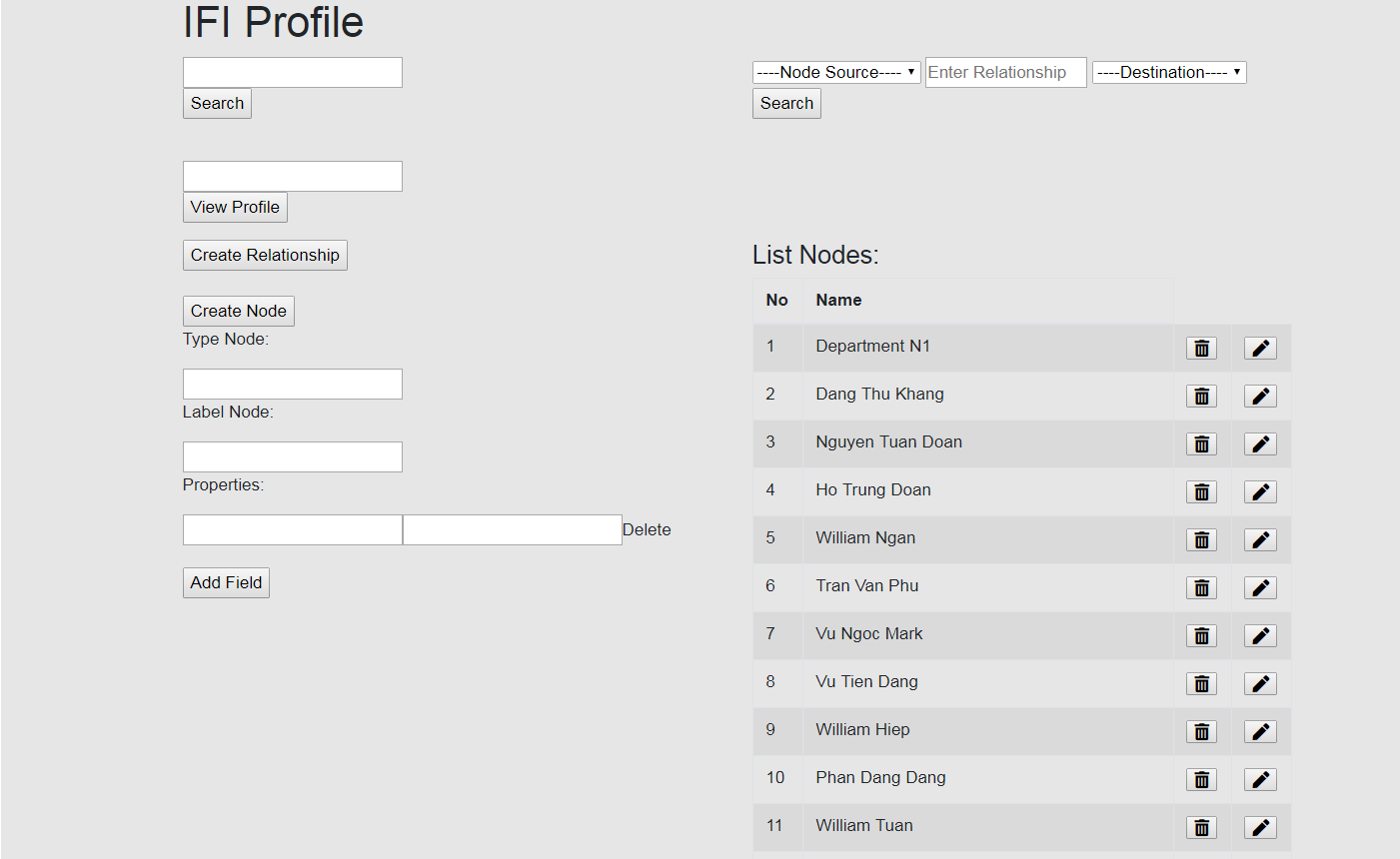
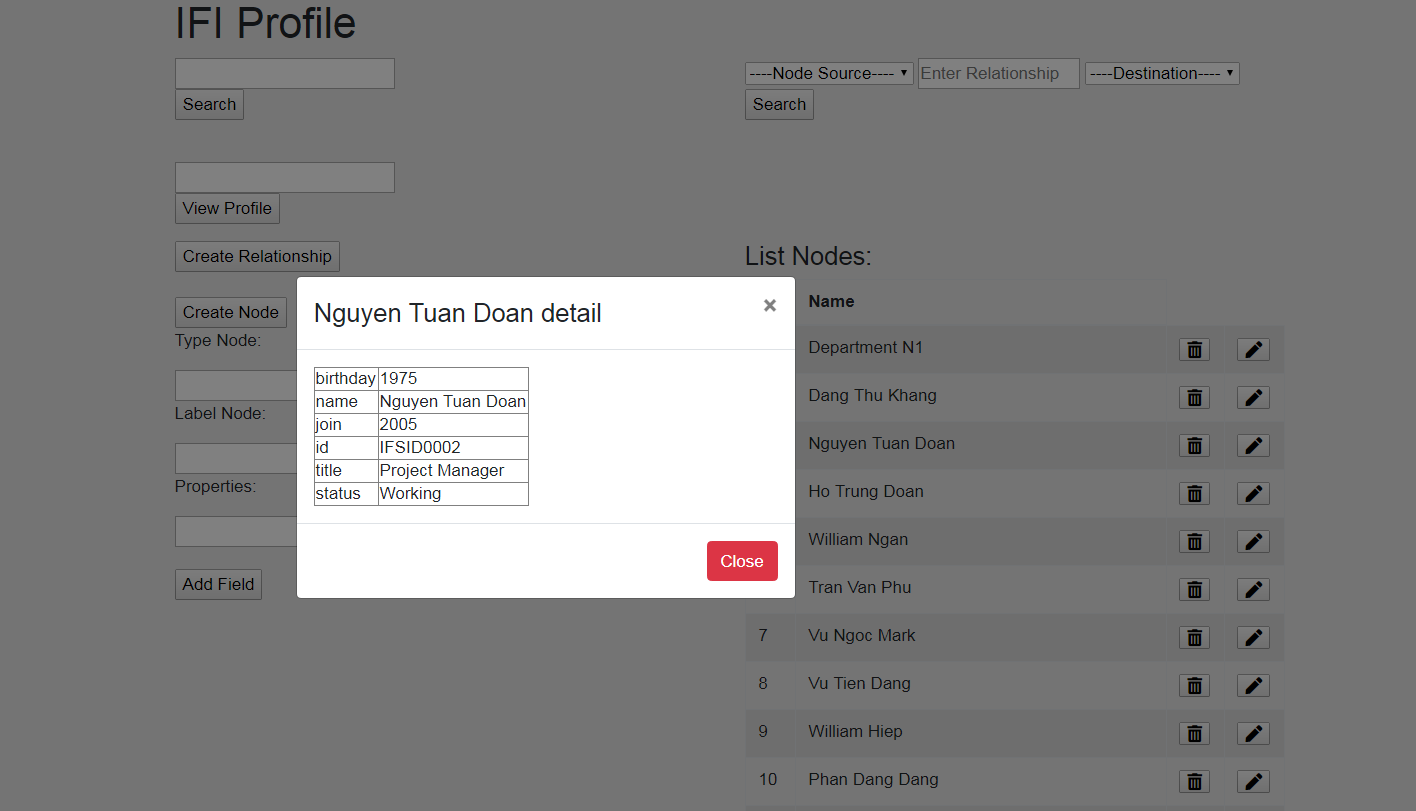
Report tổng hợp các chức năng

* Show all node to screen
* Giao diện trang home chứa list các node và các chức năng như hình:



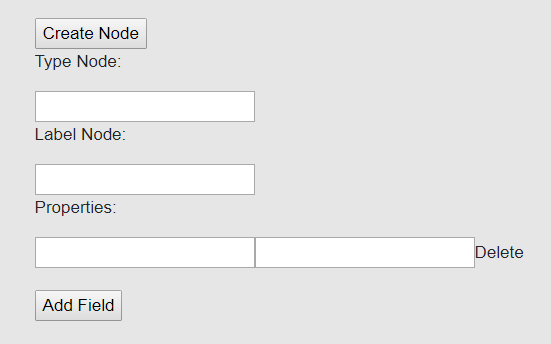
Click vào một node bất kỳ sẽ hiển thị ra thông tin cơ bản của node đó



* Cách thực hiện:

Sử dụng câu query động bằng cách cộng String để tạo ra câu query của neo4j để lấy thông tin của các node và print ra màn hình. Sử dụng mảng hai chiều để lấy thông tin riêng của từng node và Javascript để tạo event hiện thông tin ra screen.

* Create new node:
* Tạo một form như hình:

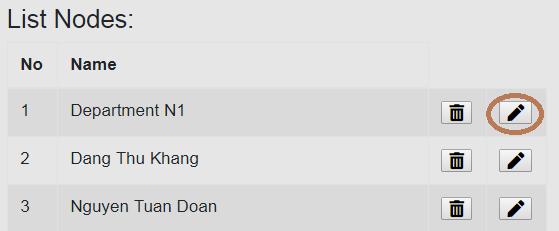


Trong đó các thông tin có thể nhập vào tùy ý, user có thể tạo bất kì node nào không có các field cụ thể. Click “Add Field” để tạo thêm các field mà user muốn, số lượng field là tùy theo user muốn tạo. Click “Create Node”, một node mới sẽ được tạo ra với tất cả các properties đã được add từ “Add Field”.

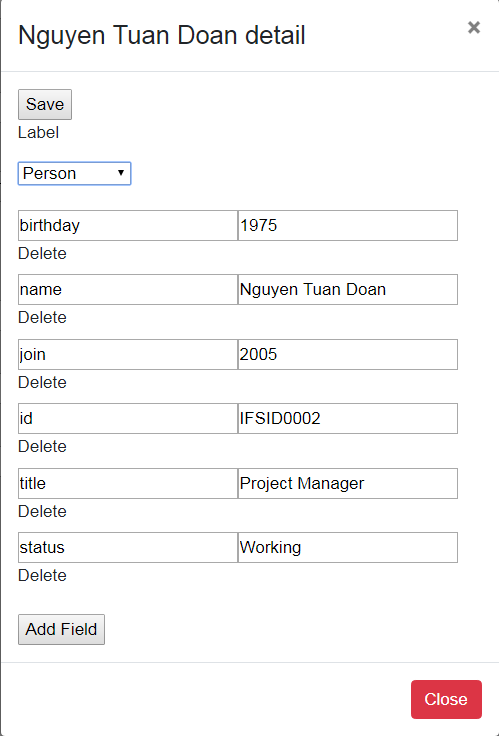
* Cách thực hiện:

Sử dụng câu query động bằng cách cộng String, sử dụng JS để tạo event cho button “Add Field”

* Update node(updated)
* Trong giao diện của home page đã show ra list các node, bên cạnh mỗi node có hai icon trong đó icon “pen” là button update. Biểu diễn như hình:



Click vào button “pen” một modal sẽ được show ra chứa thông tin của node đó. Trong đó các field được hiển thị với phần “key” là tên các field và “value” là giá trị của field đó. Các “value” có thể sửa tùy ý, các “key” là label không thể sửa. Có thể xóa một field và có thể add field mới tùy theo ý muốn của user. Tuy nhiên phần label vẫn chưa thể lấy ra tự động nên vẫn phải chọn label mới có thể chạy thành công, trong đó các label được lấy ra từ database. Ví dụ như hình:



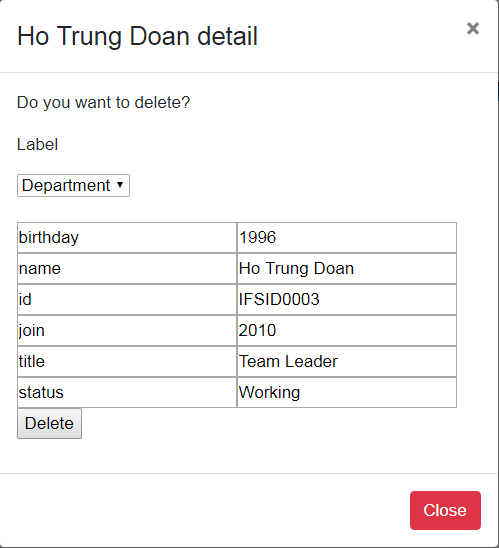
* Cách thực hiện:

Sử dụng câu query động bằng cách cộng String để lấy thông tin và update node.Sử dụng Javascript để lấy thông tin của node và show ra screen và event “Add Field”. Sau khi click “Save”, add node vào list mới và hiện thị ra màn hình.

* Delete node(updated)
* Tương tự như update node, bên cạnh node có icon “trash”:



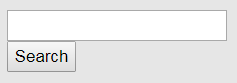
Click vào button “trash” một modal sẽ hiển thị ra thông tin của node đó nhưng khác với function update, tất cả thông tin node đều ở dạng label không thể sửa xóa từng field. “Delete node” vẫn phải chọn label như “Update node” nếu chọn sai label sẽ không thể xóa được node. Click “Delete”, node sẽ bị xóa. Ví dụ như hình bên dưới:



* Cách thực hiện:

Sử dụng câu query động bằng cách cộng String để lấy thông tin và delete node.Sử dụng Javascript để lấy thông tin của node và show ra screen. Sau khi click “Delete”, delete node khỏi list và hiện thị list mới ra màn hình.

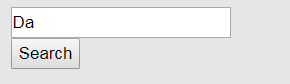
* Search node by name(updated)



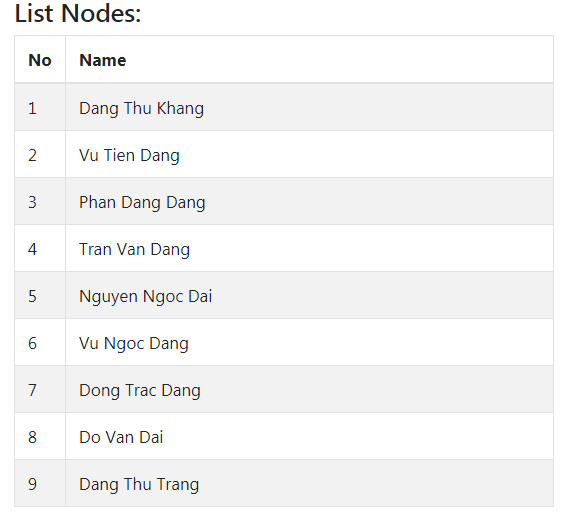
Có thể nhập bất kì chữ cái hay keyword nào nếu như có node có chứa keyword đó sẽ hiển thị tên của node ra màn hình và có thể xem được thông tin của node(keyword nhập vào phân biệt chữ in hoa và chữ thường).

Example:

Search với keyword “Da”:



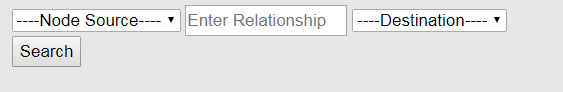
Và kết quả:



* Cách thực hiện:

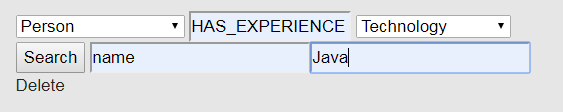
Sử dụng câu query động với ‘contains’ để bắt keyword trong database. Sau đó in danh sách các node tìm kiếm được ra một list và hiển thị list ra màn hình.

* Search node by relationship(updated)
* Giao diện được tạo như hình:

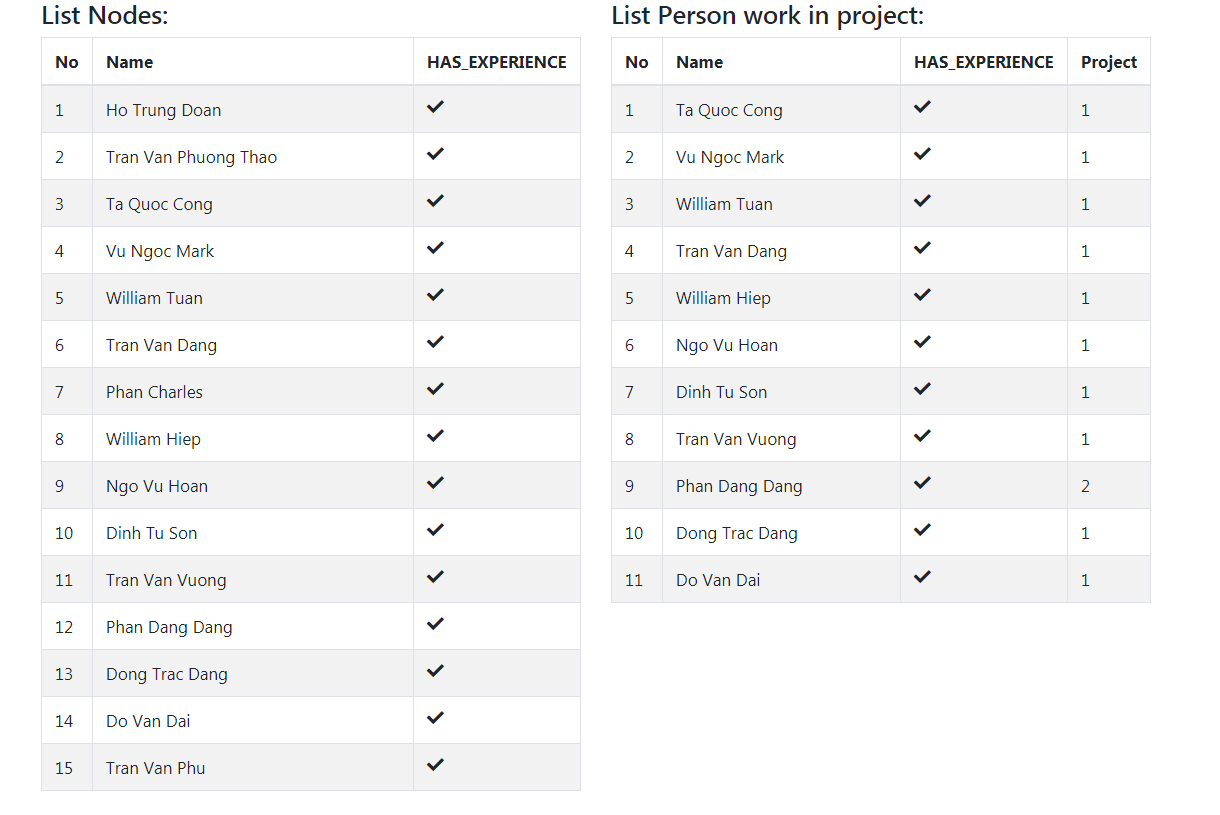


Theo thứ tự là node nguồn, relationship, và node đích. Khi click và chọn label trong node đích(destination) sẽ hiện ra một row input để nhập thông tin node đích.

Example: tìm kiếm những người có kinh nghiệm với Java



Và kết quả:



Sau khi click “Search”, kết quả sẽ trả về hai bảng trong đó, một bảng show ra số người “HAS\_EXPERIENCE” với “Java” và một bảng show ra số người vừa “HAS\_EXPERIENCE” với Java vừa làm trong dự án sử dụng “Java” và count số project người đó tham gia có sử dụng “Java”.

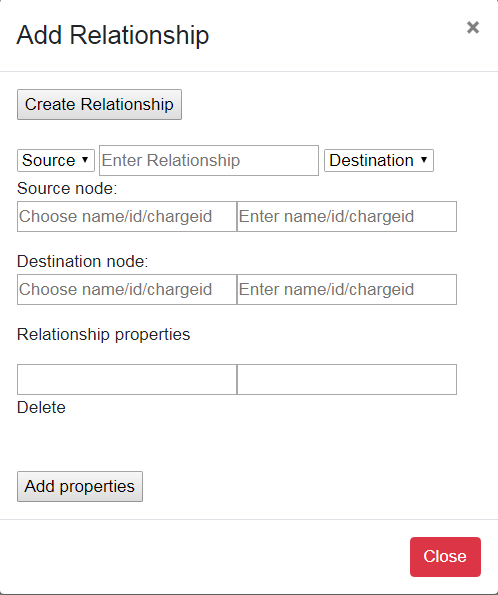
* Cách thực hiện:

Sử dụng câu query động để có thể chọn được node nguồn và đích tùy ý, sử dụng keyword ‘where type(r)’ để chọn được relationship bất kỳ. Add các node tìm kiếm được vào một list và hiển thị list ra màn hình. Sử dụng “count” để count số project mà một người tham gia

* Add Relationship(updated)
* Tạo một button “Create Relationship” như hình:

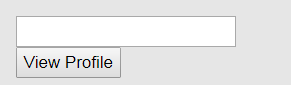


Khi click vào button, show ra một bảng có chứa form để tạo relationship:



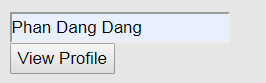
Select box là các label được lấy động từ database, Relationship cần nhập vào có thể là bất kì quan hệ nào mà user nhập vào không yêu cầu dùng relationship có sẵn trong database. Để “Create Relationship” cần nhập thông tin của node nguồn và node đích. Có thể add thêm properties cho relationship. Click “Create relationship” quan hệ giữa hai node sẽ được tạo ra trong database không hiển thị ra màn hình.

* Cách thực hiện: Sử dụng câu query động để có thể tạo relationship bất kì. Viết query để lấy hết các label trong database, gán các label vào một list và hiển thị ra ở select box. Các properties có thể tạo tùy ý, có thể có hoặc không có các properties của relationship.
* View profile:
* Tạo một input với giao diện như hình:

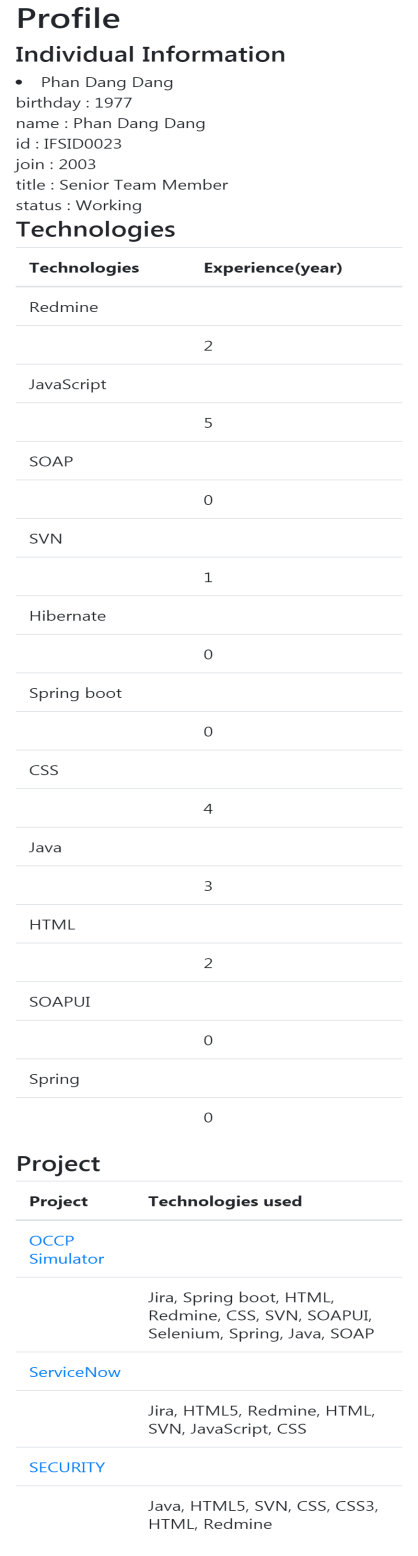


Tên nhập vào phân biệt chữ in hoa và chữ thường.

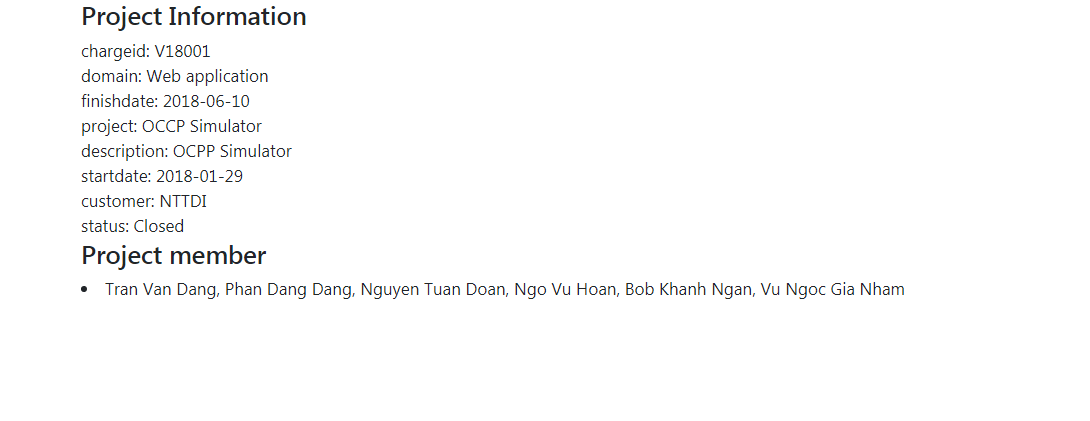
Example:



Click “View Profile” sẽ hiển thị ra một page khác bao gồm các thông tin cá nhân của Person được lưu trong node, các “Technologies” mà người đó “HAS\_EXPERIENCE” và số năm kinh nghiệm của người đó với “Technologies”, các project mà người đó tham gia và các technologies được sử dụng trong project đó:



Tại phần project, khi click vào tên project, page sẽ chuyển sang một page mới có chứa đủ thông tin của project và số người tham gia trong project đó:



* Cách thực hiện:

Sử dụng câu query động bằng cách cộng String, sử dụng hàm collect() trong neo4j để lấy ra list Technologies và list Person tương ứng với project, sau đó truyền vào list để show ra màn hình.