Lab Assignment 8

Fall Semester 2018

May 08th

1 Introduction

This eighth lab assignment will be graded and you will be working by yourself. We expect from you to not collaborates with others during the activity. Your work must conform to the *Code of Academic Ethics* included at the end of this document.

2 Description

In this assignment you will build a Ruby on Rails application from scratch, including models, controllers and views for the resources listed in the next section. In particular, you will be applying what you have learned in the past couple of weeks, in relation to the view layer of a modern web application. This means, that you will have to work with frontend code based on HTML, CSS and JavaScript, along with the backend implementation in RoR.

Your app will be a inventory management tool, providing a means for small merchants to keep track of their products. The application will be quite simplistic (and insecure), as all users will have the same role, and any user will be able to see information from all others.

3 Goals

The next list of items is what you have to accomplish by the end of the lab assignment:

1. Start a new rails project inside the GitHub repository cloned by you

```
$ cd <git repository path>
$ rails new .
```

2. [1.0pt] Generate your models, controllers and views for the resources listed below. You may use rails' `scaffold` generator command. Please use the names as listed below, otherwise, your final grade may be subject to penalties.

   - User
     - `first_name: string`
     - `last_name: string`

  - `username: string`
- Product
  - `name: string`
  - `weight: integer`
  - `price: integer`
  - `user_id: integer` (reference to user)

3. [0.5pt] Add RESTful routes for the application resources as follows: The `Product` resource must be nested within the `User` resource in the routes definition.

4. [0.25pt] Add Bootstrap to your application views. For this, you may include the `bootstrap-sass` gem in your project's `Gemfile`. Remember to issue the command `bundle install` after you add new gems to your project.

5. [1.0pt] Create a layout for the entire application based on Bootstrap. Use the Bootstrap class `container-fluid` that takes up the entirety of the space available in the viewport. The layout must have a simplistic sidebar to navigate between products and users resources paths.

6. [1.0pt] To display the list of users registered in the application, create a partial view that allows listing all the users registered in the application. For this, the partial should follow the structure of records in the `users` table. Plus, it should contain a button that triggers the delete action for a given user.

7. [0.25pt] Add the jQuery library to your project. For this, you may include the `jquery-rails` gem in your project's `Gemfile`.

8. [2.0pt] Based on Rails' scaffolding output, modify the views and controller logic so that your application uses Ajax for loading and displaying the list of products for a user. For this, the application must show the user a list of users to select from. After the user selects one of the available users in the application, the chosen user's products should be displayed. Use the `get` method provided by jQuery to retrieve a list of JSON objects from the server (e.g. your index controller action must generate a JSON response - see `http://guides.rubyonrails.org/action_controller_overview.html#rendering-xml-and-json-data`). Then add JavaScript code in the client to process the JSON response and dynamically render a table displaying the products. You may want to check the jQuery API documentation (with regard to `get()`) here: `https://api.jquery.com/jquery.get/`. In addition, here are some indications on how to use jQuery with Rails: `https://www.engineyard.com/blog/using-jquery-with-rails-how-to`. Finally, should you need to execute JavaScript inmediatly after the page loads, recall that Rails adds Turbolinks by default and this may affect the event handlers. You may either choose to deactivate Turbolinks, or handle the page load event as described here: `https://stackoverflow.com/questions/18770517/rails-4-how-to-use-document-ready-with-turbo-links` (i.e., handle the "`turbolinks:load`" event).

4 Lab setup

You should create your application inside your repository previously cloned from the Github Classroom at `https://classroom.github.com/g/exFqFRAc`. Remember to clean the git credentials set on your machine if them do not belong to you, using the following commands:

```
$ git config --global user.name "name"
$ git config --gobal user.email "email"
```

5 Grading

Your grade will be computed based on your accomplishments, considering the scores listed above for every milestone and how similar your own web page is compare to the original one.

6 Useful links

The following links to Rails Guides will provide you useful information for completing your assignment:

- `http://guides.rubyonrails.org/layouts_and_rendering.html#using-partials`
- `http://guides.rubyonrails.org/routing.html#nested-resources`

7 Code of Academy Ethics

During the course and for every task, test and homework given to you, all the ethical criteria established by the Faculty of Engineering and Applied Sciences at the Universidad de los Andes, Chile will apply:

> "Any detection of copying, plagiarism, or dishonest behavior, independent from the fact itself, will be reviewed by the Faculty Council. The minimum sanction will be a 1.0 as the final grade of the course, with the possibility of escalating to the expelling of the student from the University.
>
> Any student suspicious of unethical behavior will be punished as if he/she had committed or executed the suspected dishonest behavior. In other words, it is exclusively the responsibility of the students to behave correctly and ethically."