SECOND PROJECT

Now that you've wowed us with your class-directed Tic Tac Toe projects, it's time to show off how creative you are (all while getting practical experience with Ruby and Rails!).

The objective of this project is to:

- Practice building basic Rails apps that connect to a relational database (Postgres)
- Use RESTful routes
- Implement CRUD (Create/Read/Update/Destroy) functionality in an app
- Write an authentication system by hand (no gems besides BCrypt!)

You will be working **individually** for this first project, and you'll be designing the app yourself. Remember to keep things small -- scope creep/feature creep is the biggest pitfall for this project!

CORE REQUIREMENTS

Make sure to do all of the following with your app.

- Rails 4 app with Postgres backend
- Full CRUD functionality in at least one controller
- Authentication by-hand
- RESTful routes

CHALLENGE ADD-ONS

These are for extra credit! Don't forget -- you'll need to have your base app complete before you build any of these in -- so plan for different stages of your project, including one that meets only the core requirements.

- Style your app with Bootstrap/SASS
- Have multiple models/additional controllers
- Use an API -- Google Maps, Yelp, etc
- Integrate Angular into your application

PLANNING & DELIVERABLES

Project Plan deliverables, due Friday, Feb. 6th:

- Pitch -- sell the instructional team on your project idea (make sure that it makes sense with the objectives!)
- Stories in Pivotal Tracker -- we'll be paying closer attention this time, so be sure to keep

up with your Tracker

Completed Project deliverables, due Monday, Feb. 23rd:

- Link to Heroku hosted project
- Link to source code on GitHub

TIMELINE

- Project cycle: Weeks 4-6
- Demo day: Monday, Feb. 23rd, Week 7
- Code reviews: Week 7

Things to keep in mind

Make sure that your code is:

- DRY. Follow Ruby best practices!
- RESTful. Do things the Rails way.
- Well-formatted. Are you indenting well? Can we find the start and end of every div, curly brace, etc?
- Well-commented. Will I understand what is going on in each block or function?
- Clear. Do your naming conventions make sense? Would another developer be able to look at your app and understand what everything is?

We'll also be looking at:

- Quality of communication around decision-making. Can you defend why you chose a certain technology or why you implemented your solution in a certain way?
- Your ability to pick up new technologies and push yourself. Were you able to make the app you pitched? What makes it exceptional?