Senior Seminar Final Project Report

“Think Tank” - Arron Croft, Tyler Cabutto, Carlton Thomas

Section I. Report

* Arron Croft

In this project, we learned a lot about Ruby on Rails, ReactJS, the MVC system, and web development in general. More importantly, however, we learned to organize and efficiently operate as a team to develop software. In much the same way a job works, we operated professionally and set goals and timelines for ourselves, and I believe this project and class helped to prepare us for a future as programmers. As much as I used my skills to develop the website, I used another set of skills to organize the group and our workload. Meanwhile, my development skills were put to the test, as we strove to maintain a balance between all of our work and ensure that we supported the other member’s features and tested and resolved the many issues that arose as the features we developed interacted and we had to use our problem-solving abilities to keep the site and all of its features working every time a new feature, gem, or route was introduced to the grand scheme of the website. Overall, I think I’ve put more time into building this site than any programming project I’ve done thus far, and I thoroughly enjoyed being put to the test and creating something of this magnitude.

* Tyler Cabutto

This project involved all of us pulling together to learn new technologies like Ruby on Rails and ReactJS in order to develop a web application in a way that I have tried in the past independently but could not have done it without the guidance given from the instructor and my team. I feel as though I have over-excelled any expectation of myself that has ever occurred in any coding project I’ve ever had in the past and was honored to have such a wonderful team to work with as my final project before graduating on May 5th, 2018. This project was different and allowed me to put past time management skills I have learned in jobs while being a student here at Valdosta State University, while also learning from and showing my teammates strategies to reach a certain goal. We learned a good way to manage and delegate work load as I did my best to set deadlines for us all and ensure that I followed up with my own promises to not let the team down. This experience was a roller coaster with plenty of turns, and it is one that I would not trade in for any other type of experience. This, in my opinion, helped us get an idea of what is to come in our professional lives as we delve into the field of computer science, and find a role that suits are expertise the best. I appreciate every single one of you.

* Carlton Thomas

This project was great in that it provided us with a real life example of what to expect from clients when making a product for them and provided an incentive for competition. Overall, I would say, the single assignments helped with group assignments. I would say that this project definitely taught me a lot about database structure and how a website works in general. In the group setting if there was a problem that someone had trouble solving usually one of the other members would be able help solve that problem or at least think of an alternative. I have a better understanding of the tools used to create this final project (i.e. git, Heroku, cloud9, etc.) One improvement I would suggest is that the group project specifics should be given out earlier. This way the design and some basic idea of how to structure the data and be thought about earlier and students can be more prepared for their first group meeting.

Section II. Work Credit

* Arron Croft

- Helped group learn GitHub, and started the use of pull requests to manage commits and mergeability

- Organizing project by creating issues in GitHub and making a to-do list of them

- Developed design and visual layout of website

- Created most of the migrations for database tables (i.e. Deals, Businesses)

- Set up models to work with database, through 'has\_many' and 'belongs\_to' associations

- Added simple\_forms gem and redid all relevant forms for the site using it

- Created most of the validations for the models, specifically time-oriented ones, using 'validates\_timeliness' gem

- Set up image uploads for business and personnel profile images, as well as event images

- Set up admin dashboard using 'rails\_admin' gem, and specified which models to show and how to show them

- Split half the work of setting up Devise account system with Tyler Cabutto

- Split half the work of setting up Pundit policies with Tyler Cabutto

- Created favorites system for deals using 'acts\_as\_votable' gem

- Created RSVP system for events using 'acts\_as\_votable' gem

- Created tester database filling files in 'lib' folder using Faker gem

- Added pagination to deals and events (excluding React page) using Kaminari gem

- Set up Bootstrap throughout the site (i.e. buttons, forms, profile layout, navbar, etc.)

- Set up all 'show' pages with bootstrap to look nice and display information aesthetically

- Most of my work was done in creating the views and using Bootstrap and CSS to make the site easy to navigate and pleasant to look at

* Tyler Cabutto

- Setup team dates for when the team could meet and helped set deadlines and goals for the week and month.

- Worked on the design documents using Balsamiq.

- Sketched final design document based off feedback from clients.

- Created migrations for database tables alongside modifying them to best suite what was to be accomplished.

- Modified and created most of the forms for each type.

- Added carrierwave gem and prepared for Heroku deployment

- Added fog gem and mini magick gem to gemfile to setup.

- Modified some of the routes in the routes file.

- Created the catalog controller to allow all deals to be displayed from index.

- Created the Account table and installed and setup devise with Arron Croft.

- Split work for the pundit setup and policies with Arron Croft.

- Setup sendgrid functionality to send email once account is created.

- Created the different account types to be used in the project such as Business, Personnel, and SuperAcccount.

- Added the signup page for the accounts

- Added chartkick and installed chart.js. Also created statistics controller to manage the charts.

- Added Blazer gem to have dashboard for analytics.

- Added ahoy gem to track more analytics of visitors and other events that occur on the webpage to be visualized.

- Implemented gmaps4rails and geocoder gem with the Bing api. Added them to the appropriate pages.

- Added comments to application and tested for functionality.

- Added profanity gem to comments.

- Implemented report button and tested for functionality.

- Created seed and necessary files to load onto the database when prompted.

- Fixed bugs that were occurring towards end of development once pushing to Heroku with rails admin dashboard and blazer dashboard, among other things.

* Carlton Thomas

- Created a rough draft of the database design.

- Implemented functionality of upvotes and down voters on react page.

- Implemented functionality of favorites on the react page.

- Set up react js and webpack using the appropriate gems.

- Implemented transaction table

- Implemented dual functionality to either type or scan qr code to automatically update transaction table.

- Created load file to load Military personnels.

- Added some migration files such as fk to deals.

- Created search and sort functions for Deals on SPA.

- Created search and sort functinos for Events on static page using jQuery and JavaScript.

Section III. Pre-Loaded Accounts & Roles

We only have one account pre-loaded, which is the admin account. The other 2 account types are ‘personnel’ and ‘business’, which you can create at your discretion via the Devise sign up page.

* Account Email: super@cmfp.com

Account Password: password

Section IV. Setting Up Project

In this section, we’ll go through the commands needed to set up our project from the ground up. After setting the remote to our repository and running ‘git pull origin master’, do this…

* bundle install
* yarn install
  + Alternatively, if this fails, try ‘yarn install --ignore-engines’
* rails db:migrate
* rails runner lib/load\_deals\_and\_events.rb
* rails runner lib/create\_admin\_account.rb
* rails <db:seed>
* foreman start