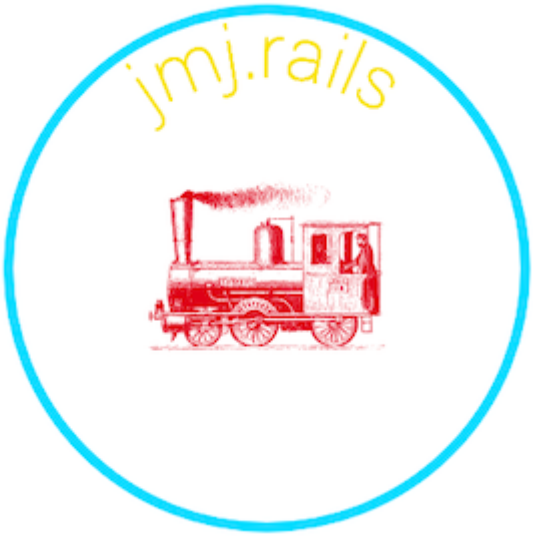
Project Report for Team `jmj.rails`

> Johnnie Oldfield, Montrez Cox, Joanne Wardell



Section I. Contributions

Montrez, Joanne, and Johnnie used built-in GitHub Project Management tools to assign tasks, implement a workflow, and gauge progess. Slack direct messaging was used to fascilliate easy communication. GitHub notifactions were integrated into Slack.

Team Member *Montrez Cox*

Montrez was majorly responsible for integrating user authoriztion via the `Pundit` gem. In addition, he added administrative dashboads via the gem `Administrate`. Approximately halfway through the project, we decided to tear down our database and rebuild due to a change in conceptual design. Montrez was responsible for rescaffolding the entire database at this point. The resulting system of tables was successful and is the cornerstone of our application. Another prime responsibility entrusted to Montrez was integrating the `ReactJS` framework into the project.

In addition to his prime responsibilities, Montrez issued fixes to the views, pundit policies, and conceptual deigns surrounding the project.

Team Member *Johnnie Oldfield*

Johnnie Oldfield's contributions were majorly involved with implementing forms. Our database involved nested associations down to two levels; moreover, this presented the need for at least two levels of nesting on forms. Johnnie located the `cacoon` gem which streamlined form-nesting associations. He implemented the form rendering and backend logic between forms and tables. Johnnie majorly contributed to the evolution of our database by creating migrations when changes needed to be made. In addition, he

authored numerous additions to the models to fascillitate associations. Also worth noting is that Johnnie initialized the rails project via rails commands at the beginning of Phase II.

In addition to his extensive work with the forms and models, Johnnie made several updates to controllers, and the wireframe design.

Team Member *Joanne Wardell*

Joanne's contributions were in configuring the production server on Heroku and developing the view pages. Joanne set up the heroku project and confitgured the production database. The database used in production was PostGres. Joanne wrote the approver page using the `ReactJS` framework. Additionally, Joanne used the `Devise` gem to implement user authentication.

In addition to her work on the views, Joanne authored migrations, tweaked the database schema, authored the seeding file, and worked in controllers.

Section II. Variable Info

The following is a table containing login information for our pre-seeded users.

Type	Email	Password	Name	Department
Payment Manager	predator@jmj.rails	password	Predator	n/a
Budget Approver	briggs@jmj.rails	password	Briggs	biology
Budget Approver	boriacho@jmj.rails	password	Boraicho	math
Budget Approver	blaze@jmj.rails	password	Blaze	computer science
Budget Approver	baraka@jmj.rails	password	Baraka	chemistry
Submitter	erron@jmj.rails	password	Erron	n/a
Submitter	ermac@jmj.rails	password	Ermac	n/a
Admin	admin@jmj.rails	password	Admin	n/a

Section III. Config Commands

Cloning the Repository

To begin, clone the git repo from our team's url and move into the directory:

```
git clone https://github.com/VSU-CS-Senior-Seminar/Group-Project-for-Cox-Oldfield-Wa

cd Group-Project-for-Cox-Oldfield-Wardell
```

Install Dependencies, Set Up Database, Launch Server

Execute the `config.sh` script which is located in the project's root directory.

`./config.sh`

Commands in Detail

Here is the `config.sh` script:

```
#!/usr/bin/env bash

echo "download gems with bundle"
bundle

echo "download yarn modules"
yarn add axios
yarn add react-currency-format

echo "set up the database"
bin/rails db:drop
bin/rails db:create
bin/rails db:migrate
bin/rails db:seed

echo "start up the server"
rails s
```