

README

Introduction

This application is for the AggieSat Laboratory. This application is to monitor workday attendance and process training verification. The application requires users to sign in with their TAMU Google Account. Members will be allowed to see their workdays, check into workdays and submit training. Project Managers will accept or reject the check ins of members. The Chief Student Leader will be able to assign users to roles as well as see all member training status. They will be able to export both attendance and training data. An email will be sent to members when their training is about to expire and when it has expired. A weekly email will be sent to the CSL listing the members with expired training.

Requirements

This code has been run and tested on:

- * Ruby - 3.1.2
- * Rails - 7.1.3
- * Ruby Gems - Listed in `Gemfile`
- * PostgreSQL - 13.37

External Deps

These are needed to run the code locally.

- * Docker - Download latest version at <https://www.docker.com/products/docker-desktop>
- * Heroku CLI - Download latest version at <https://devcenter.heroku.com/articles/heroku-cli>
- * Git - Download latest version at <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

A source code editor such as VSCode is recommended.

Documentation

For understanding the code, the Ruby documentation can be found [here](#). For understanding site specifics, see the FAQ page.

Installation

Download this code repository by using git:

<https://github.com/CSCE431-Software-Engineering/aggiesat>

Tests

An RSpec test suite is available and can be ran using:

```
`rspec spec/`
```

Execute Code

Run the following code locally in Powershell if using windows or the terminal using Linux/Mac

Navigate to your git repository

```
`docker run --rm -it --volume "$(pwd):/rails_app" -e  
DATABASE_USER=test_app -e DATABASE_PASSWORD=test_password -p 3000:3000  
dmartinez05/ruby_rails_postgresql:latest` (Replace with new docker)
```

```
`cd rails_app`
```

Install the app with the following commands

```
`bundle install  
`rails db:create  
`rails db:migrate`
```

Run the app

```
`rails server --binding:0.0.0.0`
```

The application can be seen using a browser and navigating to
<http://localhost:3000/>

Environmental Variables/Files

Google OAuth2 support requires two keys to function as intended: Client ID
and Client Secret

```
`GOOGLE_OAUTH_CLIENT_ID: 'YOUR_GOOGLE_OAUTH_CLIENT_ID_HERE'`
```

```
`GOOGLE_OAUTH_CLIENT_SECRET: 'YOUR_GOOGLE_OAUTH_CLIENT_SECRET_HERE'`
```

Deployment

The application is deployed through Heroku. The pipeline is already set up, and ownership should go to the Chief Student Leader. For information on how to transfer ownership, refer to these instructions. They are also linked on the FAQ page.

https://docs.google.com/document/d/1XUlez2OYh5SqAMJWj7mshOsOozbGyg4GTCmqsb_iEhR8/edit?usp=drive_link

This is how to create a new pipeline and deploy the app if necessary:

Setup a Heroku account: <https://signup.heroku.com/>

From the heroku dashboard select `New` -> `Create New Pipeline`

Name the pipeline, and link the respective git repo to the pipeline

Our application does not need any extra options, so select `Enable Review Apps` right away

Click `New app` under review apps, and link your test branch from your repo

Under staging app, select `Create new app` and link your main branch from your repo

To add environment variables to enable google oauth2 functionality, head over to the settings tab on the pipeline dashboard

Scroll down until `Reveal config vars`

Add both your client id and your secret id, with fields `GOOGLE_OAUTH_CLIENT_ID` and `GOOGLE_OAUTH_CLIENT_SECRET` respectively

Now once your pipeline has built the apps, select `Open app` to open the app

With the staging app, if you would like to move the app to production, click the two up and down arrows and select `Move to production`

And now your application is setup and in production mode!

CI/CD

For continuous development, we set up Heroku to automatically deploy our apps when their respective github branches are updated.

`Review app: test branch`

`Production app: main branch`

For continuous integration, we set up a Github action to run our specs, security checks, linter, etc. after every push or pull-request. This allows us to automatically ensure that our code is working as intended. Relevant info from these actions can be found in .html files located in github and in the code itself.

Components of the Github Action

- Rubocop - check for coding style
- Brakeman - Security Testing
- Rspec Testing Suite
- SimpleCov - Code Coverage

Support

Admins looking for support should first look at the application help page. Users looking for help seek out assistance from the Chief Student Leader.

Extra Help

Any further issues can be directed to Professor Wade: paulinewade@tamu.edu

References

For help with Heroku, create a ticket with Heroku Support:
<https://help.heroku.com/>

For many issues, ChatGPT is a helpful resource.