Railrota Logo

Railrota

Railrota is a full-stack volunteer rostering web application written in PHP usingLaravel. It is specifically intended for the Corris Railway volunteer base as a means of streamlining railway operations, staffing thereof, and volunteer skills and grades management.

It is the substantial programming component to Oliver Earl's CHM9360 MSc Project. It accompanies the written thesis component in this Git repository, at this time of writing.

Notable functionality includes:

- Responsive, easy-to-navigate application that functions on all screens and devices.
- Supports for custom job roles (role types), and grades (role competencies) attached to them.
- Ensures that staff can only volunteer for work that they are capable of fulfilling.
- Out of the box support for both steam and powered (diesel/electric) locomotives.
- Calendar view for easy viewing of upcoming and historic shifts, and of any vacancies they may fill.
- Support for exporting shift calendar as a downloadable PDF file.

Planned future functionality (outside the scope of the MSc deliverable, but for future FOSS revisions):

- Vue.js integration for SPA-like interface.
- API service allowing access to native mobile or desktop applications.
- Exporting shifts as Outlook, etc. appointment files.
- Email notifications on upcoming shifts and vacancies.
- Default and customisable operation templates.
- Automated rostering using simulated annealing.
- Improved code quality and reduced code duplication.

Live Instance

A live instance of the application is available for testing. It can be found at: https://www.oliverearl.co.uk/railrota

Deployment

You must meet the system environment requirements to run Laravel 5.8. These requirements can be found in the official Laravel documentation.

This project is confirmed to work under Laravel Valet and Laravel Homestead (Vagrant) for development. It should work fine under XAMPP/WAMP/MAMP or in a Docker container, but these are untested.

Academic Hand-in Specific

If you have received Railrota as an academic hand-in, you should**not need to run any installation processes** unless you run into problems.

Automated Install

An automated installer is available for Linux and macOS. Simply run`bash install.sh` after downloading or cloning the repository. It is currently untested with Windows Subsystem for Linux or other Bash implementations under Windows, so proceed with caution. For help setting up your `.env` file, please see the below section.

Manual Install

- Download or clone the repository.
- You will need Composer. Installing this globally is recommended, but it is sufficient to install the .phar file to the Railrota directory.
- Run `composer install` to fetch and install dependencies. Substitute`composer` for `php composer.phar` on this and all subsequent instructions if you are running Composer locally.
- Copy the `.env.example` file to `.env`. You can customise settings however you wish, but you must specify your database information. You can use whichever database driver that your PHP installation can support - the default is **MySQL**.
- If you wish to run the program in development mode, change the`APP_ENV` value to **local**.
- Run `php artisan key:generate` to populate your `.env` file with a new key.
- Run `php artisan config:cache` to optionally cache your configuration. This is for performance.
- Run `php artisan telescope:install` to finish installing Laravel Telescope. View the Telescope documentation for more information.
- Migrate information into the database by running `php artisan migrate` (or `php artisan migrate:fresh` if you've run it previously) to provide a fresh database structure to your specified database.
- Create an admin user that you can log into using`php artisan db:seed --class=AdminSeeder`. By default, this uses **jane.doe@railrota.com** for its email address, and**password** for its password.
- Additional dummy data can be inserted into the database by using`php artisan db:seed` but this is generally unnecessary.
- You will need Node.js to use Laravel Mix and compile front-end assets, such as Sass and JavaScript.

  Install dependencies using **Yarn** or **NPM** and use either `run dev` to produce uncompiled assets, or `run production` for minified, transpiled assets. If you have received this project in an academic hand-in, the files have been precompiled for you.

- Redump the autoloader using `composer dump-autoload`.
- Have fun!

Directories

A quick run-down of the directories is as follows:

- **app** contains Railrota code, such as policies, providers, middleware, controllers, and models.
- **bootstrap** contains Laravel code used for bootstrapping the application.
- **config** contains Laravel and library configuration files.
- **database** contains model factories, database migrations, and seeding classes.
- **docs** contains PHPDoc generated documentation by Sami.
- **node_modules** contains Node.js dependencies relied upon by Laravel Mix.
- **public** is the web-facing directory, containing the application entry-point, compiled assets, and social media images.
- **routes** contains lists of all the application routes.
- **resources** contains Laravel Blade views that make up the presentation of the website, as well as styling and front-end JavaScript.
- **storage** contains log files generated by the application, and generated PDF files downloaded by users.
- **tests** contains feature and unit tests.
- **vendor** contains PHP dependencies relied upon by Laravel, as well as libraries.

Testing

The application includes a full suite of unit and feature tests written usingPHPUnit.

You can run PHPUnit tests by running `composer run test` to print out results to stdout. Alternatively, you can run `composer run test-export` to export test results to `test.html` in the root directory.

Documentation

Railrota uses Sami to generate documentation from source code docblocks.

You can run Sami by running `composer run doc` to generate a new **docs** subdirectory.

Troubleshooting

If you are having any problems, it would be advisable to check the Laravel documentation first to see if you are running into compatibility issues.

Alternatively, email me on ole4@aber.ac.uk for prompt help.

If you require assistance after the 27th of September 2019, when I will cease being a student at Aberystwyth University, please email me instead at oliver.earl@mail.bcu.ac.uk as I will be a full-time student at Birmingham City University from that point onwards.

Contributing

Contributing information will become available once the project has been made public.

License

MIT