rmit-softeng-2017

RMIT Software Engineering Process and Tools in Semester 1, 2017 Assignment repo.

Team Members

- Matt Hayward (s3537899)
- Callan Delbridge (s3537905)

Contribution Overview

- Matt Hayward (50%)
- Primarily developed the backend code and infrastructure and worked on some of the frontend.
- Callan Delbridge (50%)
- Primarily developed the frontend and worked on some of the backend.

Tutor

Lawrence Cavedon (Tuesday 9:30)

GitHub Repository

https://github.com/zcallan/rmit-softeng-2017

Running the application

Running the application is easy using the provided Docker Compose file. Simple run `docker-compose up` in the root directory of the project.

Docker installation instructions

The application will then be accessible in your browser at `http://localhost:3000`

If you have made any changes to any of the files you will have to run `docker-compose build` to rebuild the docker image.

Logging into the application

Default admin credentials are located in the server/config/users.json file.

Application file structure

__mocks__/ - Stores any mock definitions for the Jest testing suite. build/ - Stores the built front-end application and also is where static files like images and the index.html file are stored.

client/ - Stores all of the code for the client.

client/config/ - Stores all the client configuration.

client/styles/ - Stores all the styling for the client.

client/utils/ - Any common utility classes for the client are stored here.

client/views/ - All of the client views are stored here.

documentation/ - Any additional documentation for the application is stored here.

server/ - Stores all of the backend code for the application.

server/config/ - Stores all of the server configuration.

server/models/ - Stores all the models for the application.

server/mongo/ - Stores any scripts relating to the running of the database.

server/public/ - Stores any public static files that are served up by the server.

server/routes/ - Stores all the servers API handlers.

server/utils/ - Any common utility classes for the server are stored here.

Design pattern

The application was designed based on the MVC (Model, View, Controller) design pattern

Installing Swagger UI

To install swagger ui please pull the Git submodules by running the following command

```
git submodule update --init --recursive
```

Swagger will then be available to access at http://localhost:3001/api-docs

Accessing Swagger UI docs

Once the Swagger UI has been installed, you can access the API documentation at http://localhost:3001/api-docs

Running Tests

Tests can be run with npm run test.

Technologies

The following open source technologies are used:

- Docker
- Node JS
- Swagger
- React
- Redux
- React Router
- MongoDB
- Mongoose
- Webpack
- Express
- Babel
- JSON Web Token
- EsLint

Full list of dependencies is available in package.json file.