ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Surf Club Management Application

Authors: Bernardo Fragoso
Gonçalo Albuquerque
Miguel Sousa

Advisors: Filipe Freitas
Miguel Pires, ESC

Project info carried out under the Project and Seminar
Computer Science and Computer Engineering Bachelor's degree

July 2022

Introduction

In the context of project and seminar a web application to manage surf clubs. This application is available in a Github repository to which is described in the following chapters.

Project Structure

The project is in a Github repository, with the following **URI**:
`https://github.com/BernardoFMF/surf-management-app`

The repository contains, in addition to the project's source code, the report, the description of the organization, the SQL scripts and other auxiliary documents. The documentation of the server, client and database can be found in the Wiki section of the above mentioned repository.

- **Backend** - The server application is included in this folder.
- **Frontend** - The client application is included in this folder.
- **Docs** - Database scripts, report and every other document about the project is included in this folder.

Group Constitution

The group is constituted by the students:

- Bernardo Fragoso, nº 47203 (A47203@alunos.isel.pt)
- Gonçalo de Albuquerque, nº 47265 (A47265@alunos.isel.pt)
- Miguel Sousa, nº 47270 (A47270@alunos.isel.pt)

Advisors

The group is advised by:

- Filipe Freitas (ffreitas@cc.isel.ipl.pt)
- Miguel Pires (miguel.toscano.pires@gmail.com)

Instructions To Run

1. Clone the repository:
    - *git clone* `https://github.com/BernardoFMF/surf-management-app.git`
2. Instalation of PostgreSQL.
3. Creation of a Server on PostgreSQL and one database.
4. Run the scripts present on the script folder in the following order:
    - create.sql
    - Triggers.sql
    - procedures.sql
    - Insert.sql
5. Install the dependencies (must have NodeJS installed) on the main and frontend directories:
    - *npm install* on the command prompt.
6. Create a *.env* file in the root directory with the following variables (must follow the pattern key=value):
    - PORT_NUMBER - represents the port of the database;
    - PG_USER - represents the user of the database;
    - PG_PASSWORD - represents the password of the database;
    - PG_HOST - represents the host of the database;
    - PG_PORT - represents the port of the database;
    - PG_DB - represents the name of the database;
    - NODE_ENV - must be set to the value **development**;
    - EMAIL - represents the email which will send notifications;
    - EMAIL_PASSWORD - represents the password for the email above;
    - SECRET - represents the secret string which will be used to sign the session cookie.

7. Start the application using the following command in the main directory:
    - *npm run dev* on the command prompt.
8. Here are some dummies that you can use to login:
    - **Admin**:
        - username: afonsoribeiro
        - password: 123
    - **User member**:
        - username: joselopes
        - password: 123
    - **Company member**:
        - username: ripcurl
        - password: 123

Instructions To Run The Tests

1. Execute all steps in the chapter above up to point 6.
2. Create two new databases within the PostgreSQL server previously made. This is due to having unit tests and integration tests for the database and both suites running concurrently generates consistency issues. Keep in mind all the databases must have the same credentials, differentiated only by the database name, which can be whatever name wanted.
3. In the *.env* file created, besides inserting the variables mentioned, add the following:
   - PG_DB_TEST - represents the name of the database used in unit tests;
   - PG_DB_TEST_INTEGRATION - represents the name of the database used in integration tests.
4. Start the application using the following command in the main directory:
   - *npm test* on the command prompt.