Web engeeniring project

Mercurio Verde Hopson, e10391

Index

For these project I chose the option of the streets of Braga, in this report I will go through some of the specs of the project and how did I manage to make it

Database Model and Structure

The database was made using mongo DB and is divided in three models:

- Users
- Streets
- Houses

The user model, which is used in order to create the mongoose auth has the following features:

- email: String with the email of the user
- password: String with the password of the user
- name: String with the name of the user
- type: String with the type of user, can be either Normal or Moderator

The Streets model is used to describe each street of braga, and has the following features:

- id: String with the Id of the street
- nome: String with the name of the street
- figura: List of objects, which represents the images of the street, each object is composed of:
    - imagem: name of the image.
    - legenda: legend of the image, describing it
- comments: List of objects, that represents the comments of the street, each comment has the following structure:
    - text: text of the comment
    - data: dates, years or period of time highlighted of the comment
    - lugar: place(s) highlighted of the comment
    - entidade: entity(ies) highlighted of the comment
- listaCasas: List of objects, that represents the house list of the street, each house of the list has the following structure:
    - id: id of the house

    - numero: number of the house

      This list is used to connect the streets and house models

The Houses model is used to represent each house of each street of the dataset, it has the following features:

- id: String with the id of the house
- numero: Int with the number of the house
- enfiteuta: String with the leaseholder of the house
- foro: String with the forum of the house
- rua: String with the id of the street where the house belongs to
- desc: List of objects, that represents the comments of the house, each comment has the following structure:
    - text: text of the comment
    - data: dates, years or period of time highlighted of the comment
    - lugar: place(s) highlighted of the comment
    - entidade: entity(ies) highlighted of the comment

User Auth

For the user authentication I used passport-local mongoose, in order to save the user records in a mongoose db, the structure of the auth module is the same one we used in class.

Back end

In the back end all the operations regarding the streets and houses model are taked care of, the back end is divided in:

- Model: Where the two mongoose model are defined
- Controller: Where the operations of finding, updating, deleting and added are defined for each model
- Index: Where each controller operation previously defined is associated to a url, and where the data obtained is previously prepared, so the controller does not have to do it.

Thanks to the back end the following operations are supported:

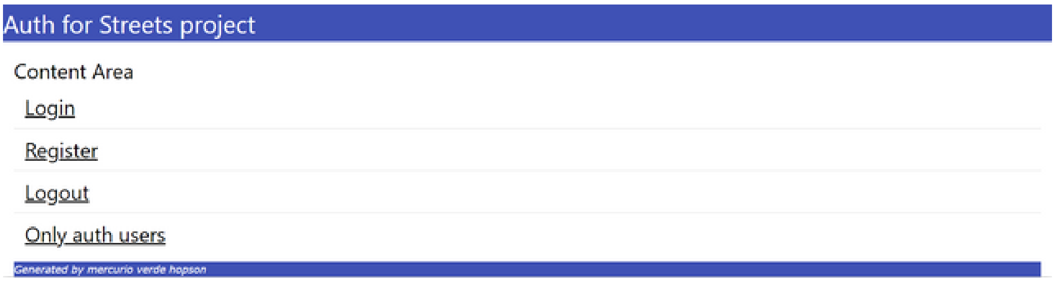
- get streets: We can fetch 6 streets at a time to then show them on the web
- search street: We can search a street by name
- filter streets: We can filter streets by the highlighted terms on their comments
- get street: Search a street using its id
- add street: Add a new street to the database
- get a house of a street
- get a comment of a street
- add a comment
- add a house
- delete a comment
- delete a comment of a house
- add a photo
- add a comment to a house
- update a comment

- update a house
- update a comment of a house

Front end(Interface)

In the interface is located all the views for the actions are found, also we have an index for both the users administration and streets administration, in the user we can login and register, and in the streets we can perform operations such as adding a street, adding a photo to a street, adding/deleting/updating a comment and adding/deleting/updating a house to a street, when performing an operation the data is fetched from the request, modified and sent to the back end. Here are some pictures of how the front end of the application looks like:

- User auth index:

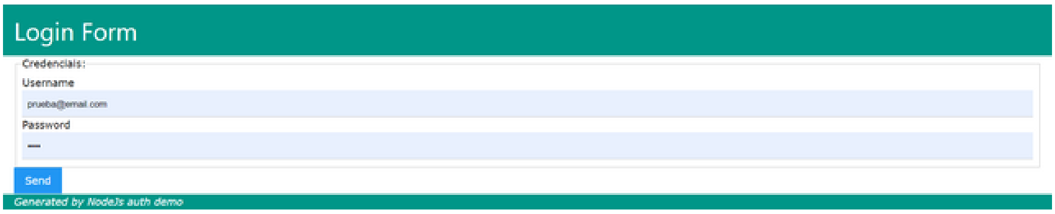


Auth for Streets project

Content Area

Generated by mercurio verde hopson

- User Log in form:



Login Form

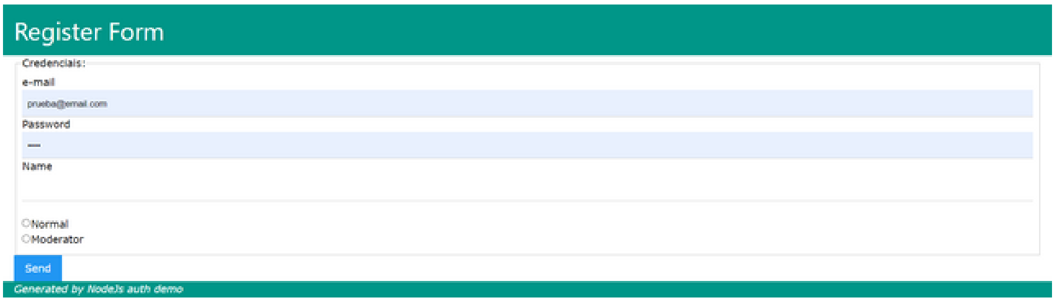
Credencials:

Username: prueba@email.com

Password: —

Generated by NodeJs auth demo

- User Register form:



Register Form

Credencials:

e-mail: prueba@email.com

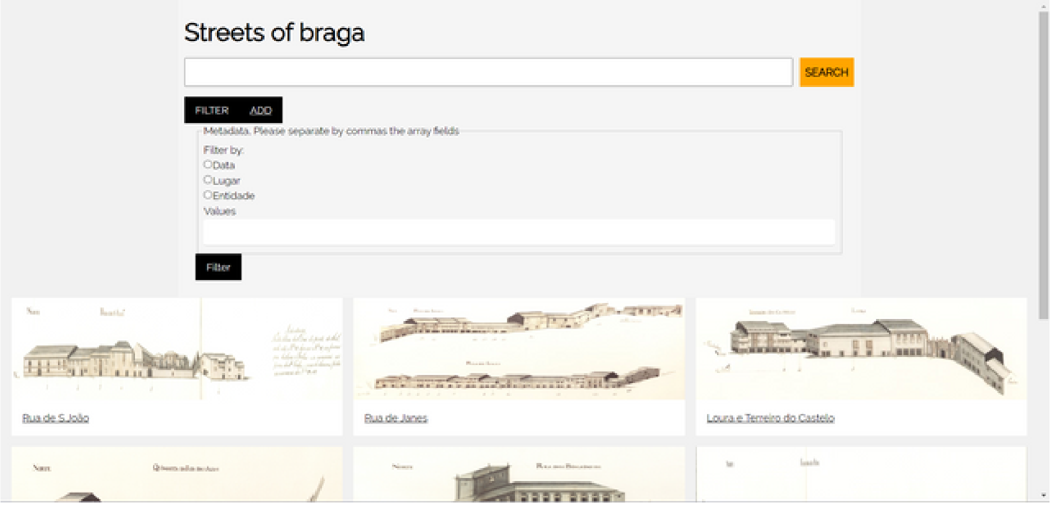
Password: —

Name

○Normal
○Moderator

Generated by NodeJs auth demo

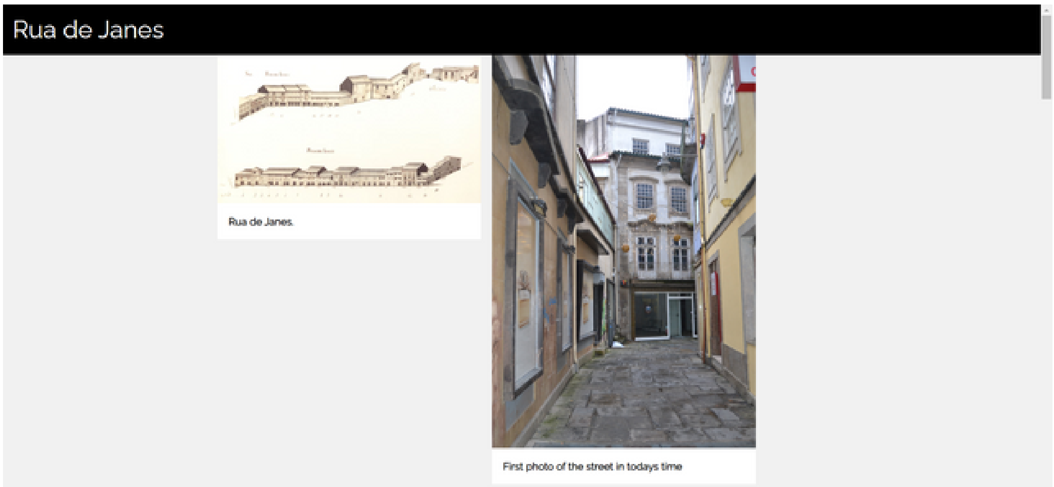
- Main window:



Metadata. Please separate by commas the array fields

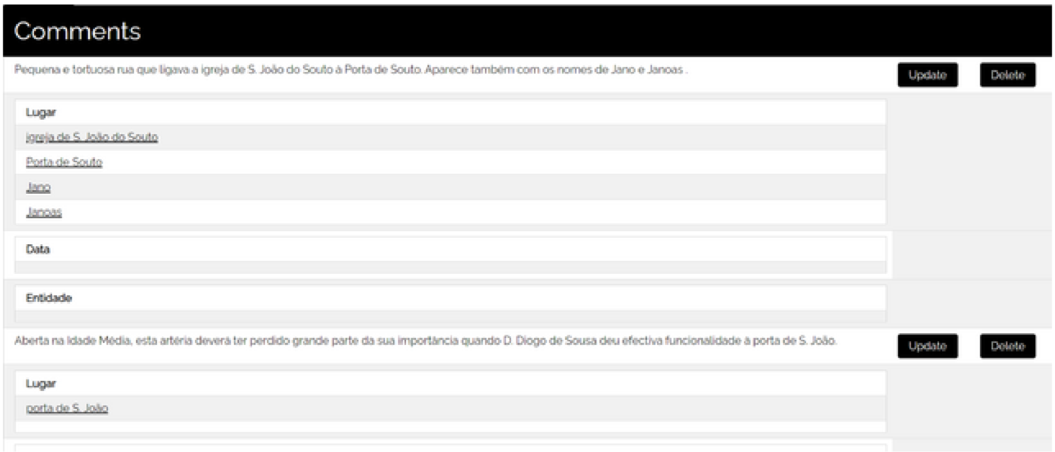
Filter by:
○Data
○Lugar
○Entidade
Values

- View street(Normal user doesnt have the options to update nor delete elements):



Rua de Janes.

First photo of the street in todays time



Comments

Pequena e tortuosa rua que ligava a igreja de S. João do Souto à Porta de Souto. Aparece também com os nomes de Jano e Janoas .

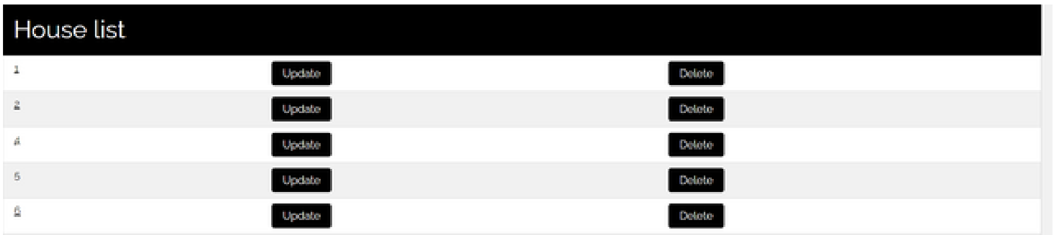
Lugar
igreja de S. João do Souto
Porta de Souto
Jano
Janoas

Data

Entidade

Aberta na Idade Média, esta artéria deverá ter perdido grande parte da sua importância quando D. Diogo de Sousa deu efectiva funcionalidade à porta de S. João.

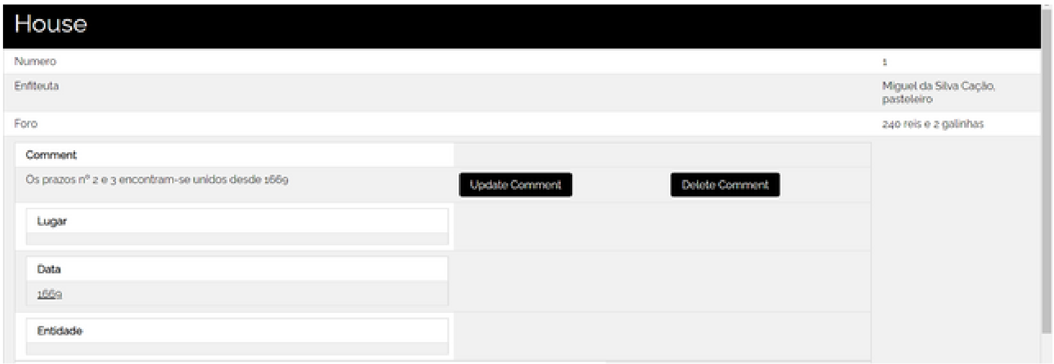
Lugar
porta de S. João



House list

1	Update	Delete
2	Update	Delete
4	Update	Delete
5	Update	Delete
6	Update	Delete

- View House(Normal user doesnt have the options to update nor delete elements):



House

Numero	1
Enfiteuta	Miguel da Silva Cação, pasteleiro
Foro	240 reis e 2 galinhas

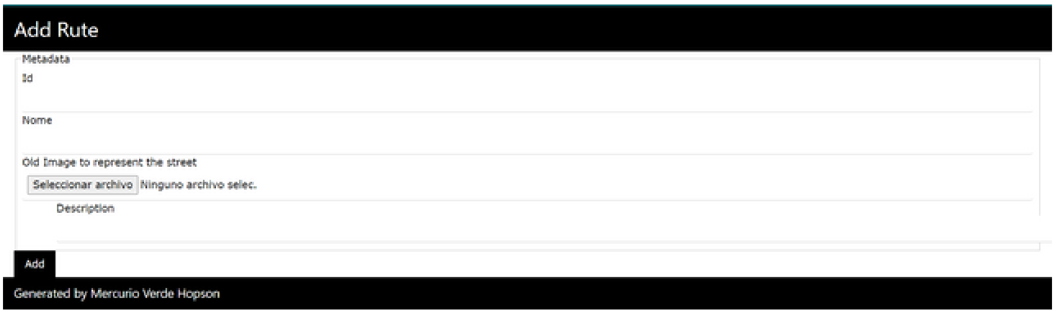
Comment
Os prazos nº 2 e 3 encontram-se unidos desde 1669

Lugar

Data
1669

Entidade

- Add Street:



Add Rute

Metadata
Id

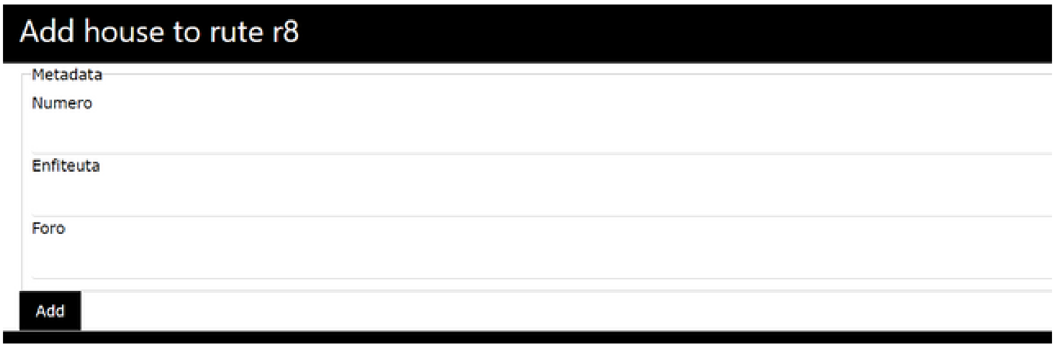
Nome

Old Image to represent the street
Ninguno archivo selec.

Description

Generated by Mercurio Verde Hopson

- Add House:



Add house to rute r8

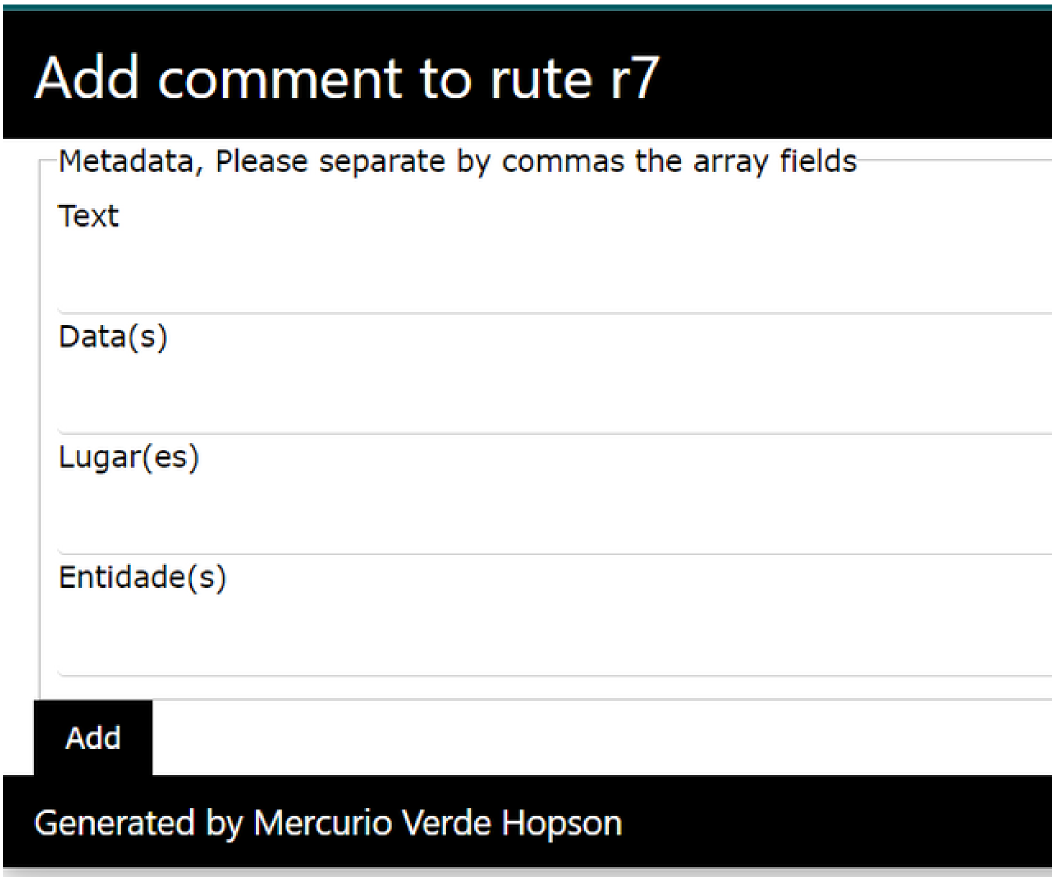
Metadata

Numero

Enfiteuta

Foro

- Add Comment:



Add comment to rute r7

Metadata, Please separate by commas the array fields

Text

Data(s)

Lugar(es)

Entidade(s)

Generated by Mercurio Verde Hopson

How it works

To run the application first the back end has to be initialized, going to the folder where is located and running npm start, then the same has to be done with the interface app, in here both the auth and the interface parts are implemented. From a user point of view, he is first introduced to the user page, where he can eather login or registered, once logged in he will find the main page, where a search bar, to find streets by their name, is found, also a filter form, that can be hided pressing the filter button, a add button, and six streets, presented by a picture and their name.

If the user decides to press the filter button, only streets that have the dates, places or entities highlighted in their comments will be shown, if he decides to press the add button he would be redirected to the add street form where he has to introduce the name and an image of the street, if he clicks on to a street, he will see all the pictures of the street(4 max), the comments and a list of the houses, if the user is a moderator he will be able to add pictures, modify or delete comments/houses, if it is a normal user he would only be able to add comments and houses. if he clicks on adding a comment or house he would be redirected to the addComment/house view, if he clicks on the house number, he would view all the house charactheristics and comments being able to add a new comment and update or delete comments if it is a moderator.