Translated from Spanish to English - www.onlinedoctranslator.com

PROJECT DESCRIPTION

We want to create an application so that any registered business can upload its own website with content (photos, text, etc.), so that it can be viewed when users search for businesses in a city and a specific activity.

PROJECT SCOPE

A front-end will be created implementing all the functionality proposed below.
(Backend will be implemented in 2Q)

TECHNICAL IMPLEMENTATION

For front-end development, NextJS will be used with calls to the backend with Fetch.

DETAILS

There will be four types of clients: admins, businesses, anonymous users and registered users.

From a back-office page,**interface for admins**, a business can be registered (by an admin) by sending the following information:

- Name of the business.
- CIF
- Address
- Email
- Telephone contact

Create admin login page and create the previous form to register businesses that, for now, we will save on the server's local disk (see note 4 of Technical requirements, at the end of this document). But implement POST request with fetch for sending to future backend.

Additionally, from the admin interface (back-office), you can consult the businesses by filtering by name and/or city. A business may also be eliminated.

Once registered, the business will be given access with a page [id] with which it can upload its content (in 2Q we will see authentication and JWT) from its**trading interface**:

- City
- Activity
- Qualification
- Summary
- [Texts]
- [Photos]
- Data not modifiable by the trade
    - Scoring
    - Number of scores
    - Reviews

Create business login page and create form that, for now, we will leave on the server's local disk. But implement POST request with fetch for sending to future backend.

Furthermore, from its interface, the business will be able to modify its data, upload text and photos (only the file name), see users in its city interested in its activity. And you can also delete your own page.

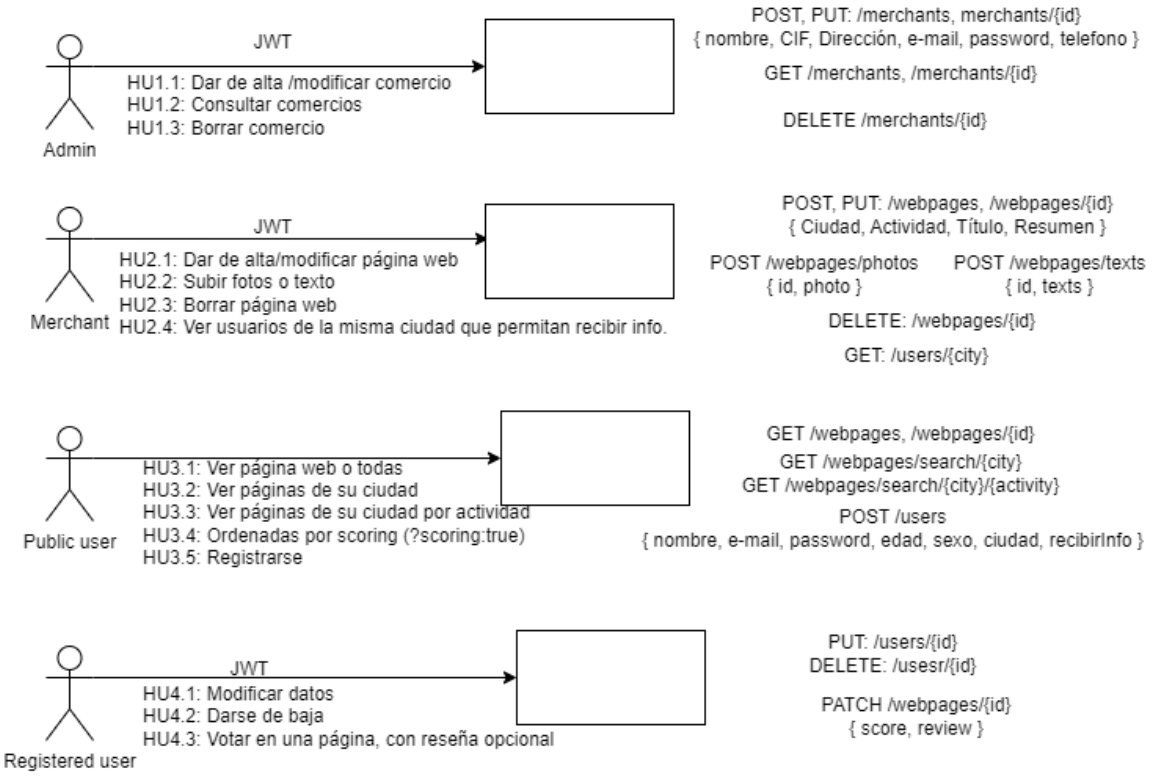
The user can be anonymous (queries) or be registered to receive offers from their city. From**user interface**You can register such that:

- Name
- Email
- Password
- Age
- City
- Interest
- AllowsReceiveOffers

Create user registration form that, for now, we will leave on the server's local disk. But implement POST request with fetch for sending to future backend.

From the user interface you can make inquiries about businesses filtered by city, activity, etc. And if the user is registered, it will allow them to rate the business and leave a review.

USER STORIES



FUNCTIONAL REQUIREMENTS

- Admins:
  - Access to the administration page through login and password
  - It is the only profile that a business can register. Once registered, you must provide credentials.
  - You can consult and filter businesses by name and city
  - They can eliminate businesses

- Shops
  - They can only upload and update content on their own page. To do this they must use credentials.
  - They cannot modify the Scoring, the number of votes or the reviews.
  - They can unsubscribe and delete their own page.
  - You can check the interests of users in your city to send them an email. For this, given a query, you will receive emails from all the users in your city interested in a specific topic and who have the “allowReceiveOffers” field set to true.

- Anonymous user
    - You can consult:
        - the shops of a city,
        - the shops of a city and an activity.
        - a specific business based on its identifier
    - You can register (no longer anonymous)

- Registered user
    - You can update your details (city, interests)
    - You can activate and deactivate the flagreceiveOffers
    - You can write reviews on a business
    - You can unsubscribe

INTERFACES (pages)

- Administrator:
    - login page
    - registration of businesses
    - shop search
- Shops
    - login page
    - consultation of user data (who have given their approval)
    - modification of data on your business page
- Unregistered users
    - shop search
    - information about a business
- registered users
    - login page
    - preferences page
    - shop search
    - commerce page with editing reviews and votes

TECHNICAL REQUIREMENTS

1.- The Web is required to be responsive, and the design is open (custom css, bootstrap, tailwind, ...).

2.- You can create the pages and functionalities with Next that you require.

3.- Functional and/or technical improvements to the proposals will be highly valued.

4.- Until we program a backend in 2Q, we will store the data on the server's local disk. Example:

```
import{NextResponse}desde'next/server'
import{readFileSync,writeFileSync}desde'fs';

exportasync functionPOST(request) {
    constdata=awaitrequest.json()
    try{// Read user.txt from disk and concatenate with data from request
        constusers=JSON.parse(readFileSync("users.txt"))
        writeFileSync("users.txt",JSON.stringify([...users,data]))
    }catch(and){// If user.txt file does not exist, create it with data from request
        writeFileSync("users.txt",JSON.stringify([data]))
    }
    returnNextResponse.json({
        messages:"Saving data..."
    })
}
```