10-5-2024

NoteTopia Documentation

Notion-like Web App using: React, Express and NodeJS

ENRIQUE SÁNCHEZ-MIGALLÓN OCHOA
FELIPE ALCÁZAR GÓMEZ
ESI – UCLM – CIUDAD REAL

Contents

1. Introduction.

Our main goal for this project is to create a **Web App** like one that most of us already know, Notion. This Web App focuses on the creation and sharing of notes across users. Based on this, we have decided to use three different **technologies** that will help us achieve our goal:

- React
- Express
- NodeJS

We can develop all functionalities required by means of **Linux OS** and **VS Code**. This environment facilitates the development of the project, allowing us to be much more productive.

Once we have defined our main goal and how it will be achieved, we can indicate a set of **requirements** that the resultant work must have, these are the CRUD operations of:

- Notes (Create, Edit, Delete and Share). Notes can have text, lists and pictures.
- Note collections (Create, Add and Remove Notes, Delete and Share).
- Users (Sign-in, Login, Logout, Delete).
- Users' friendships (Requests, Allowance, Denial, Revoke).

Furthermore, a **user management Module** must be included:

- **Administrator**: Can manage users, relationships, note collections and notes.
- **User**: Can create notes (with or without pictures) and collections, can list all his friends' notes and collections, can request a friendship relationship, and can allow and deny a friendship relationship. Also, they can share notes.

2. Diagrams.

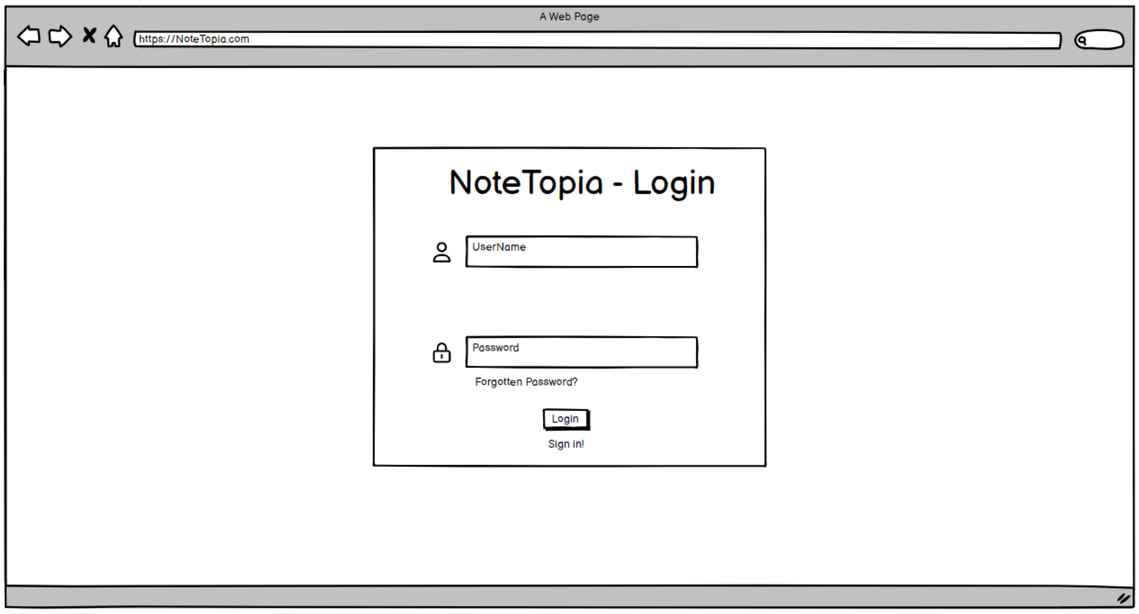
In the following section, we are going to indicate a series of diagrams that will represent the functionalities of the Web App in a broad approach.

2.1. Contents Diagram.

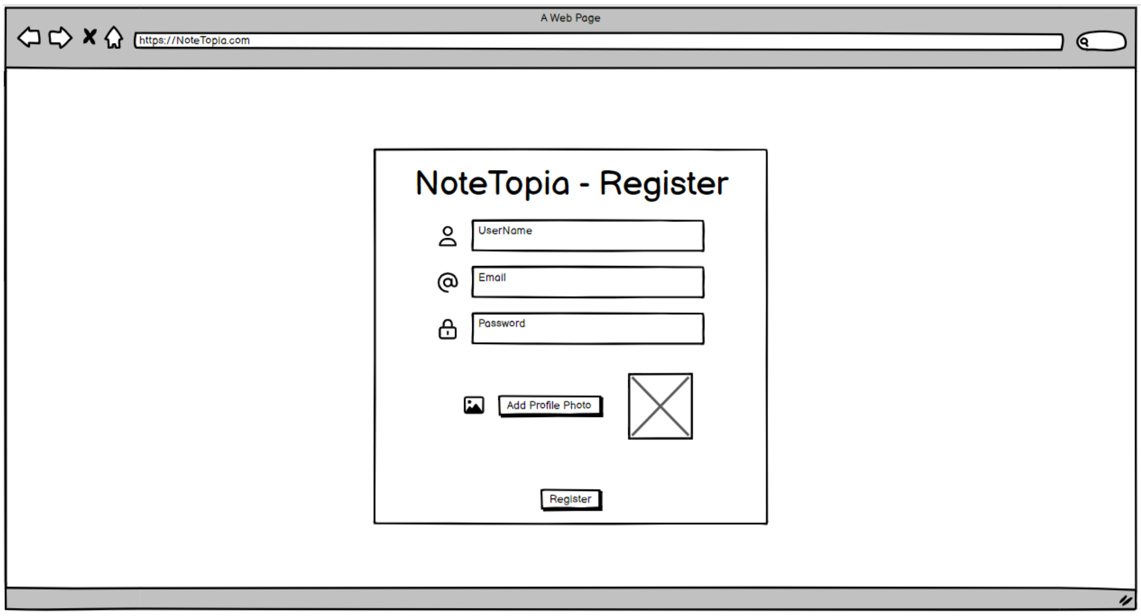
2.1.1. Mock-up's.

In the following section we can see the Mock-up's that we design of our application.

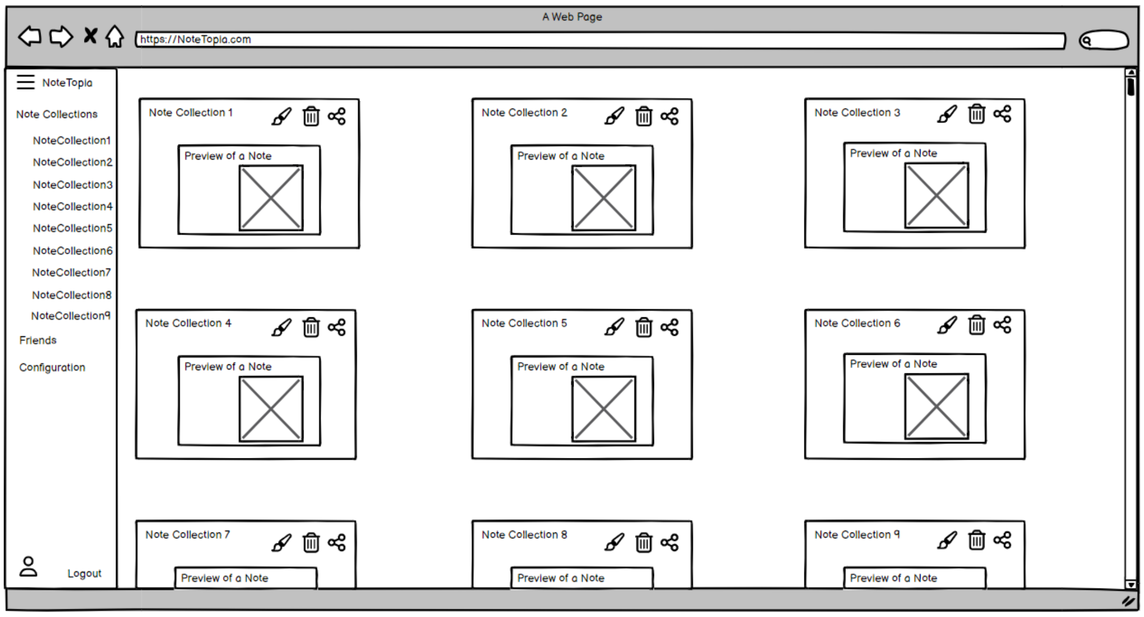
- **Login Page.**



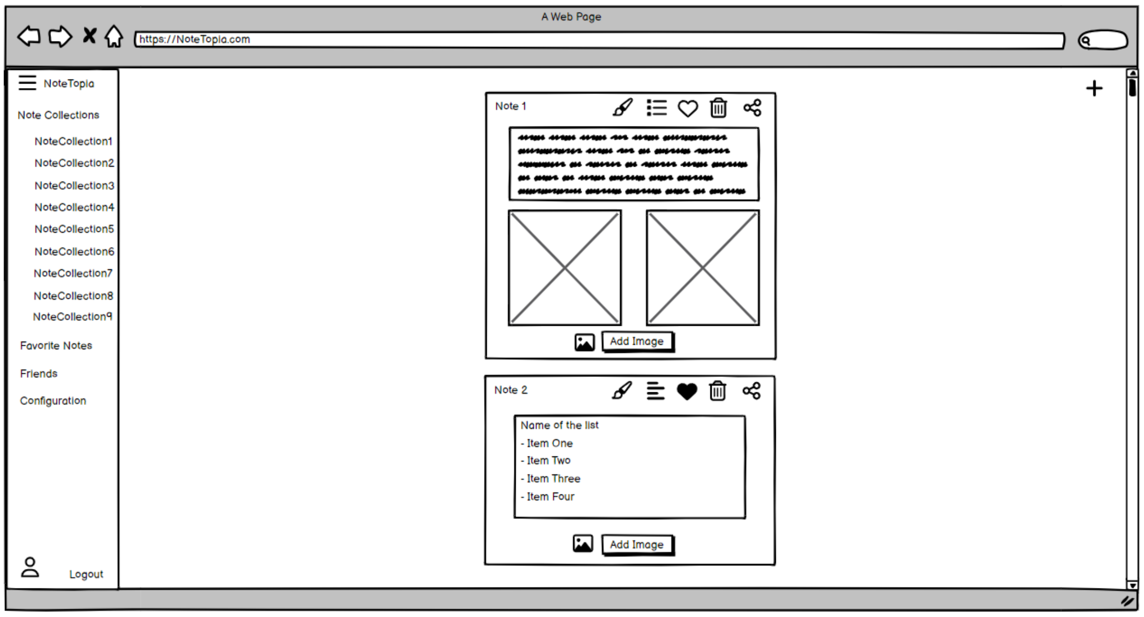
- **Register Page.**



- ## Note Collections Page.

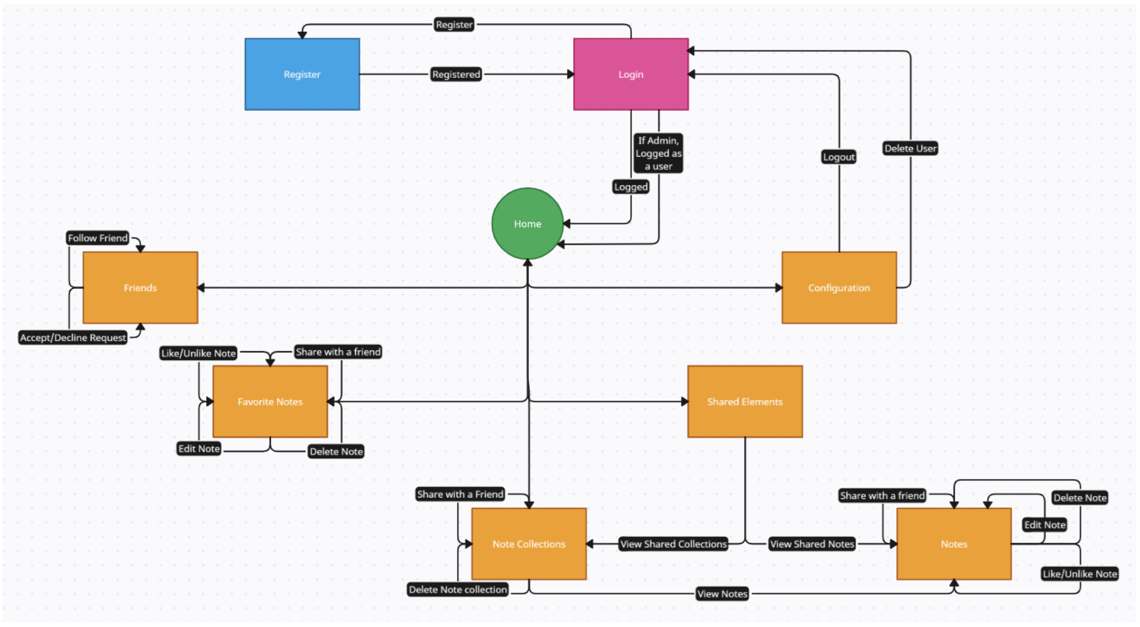


- ## Notes of a Note Collection Page



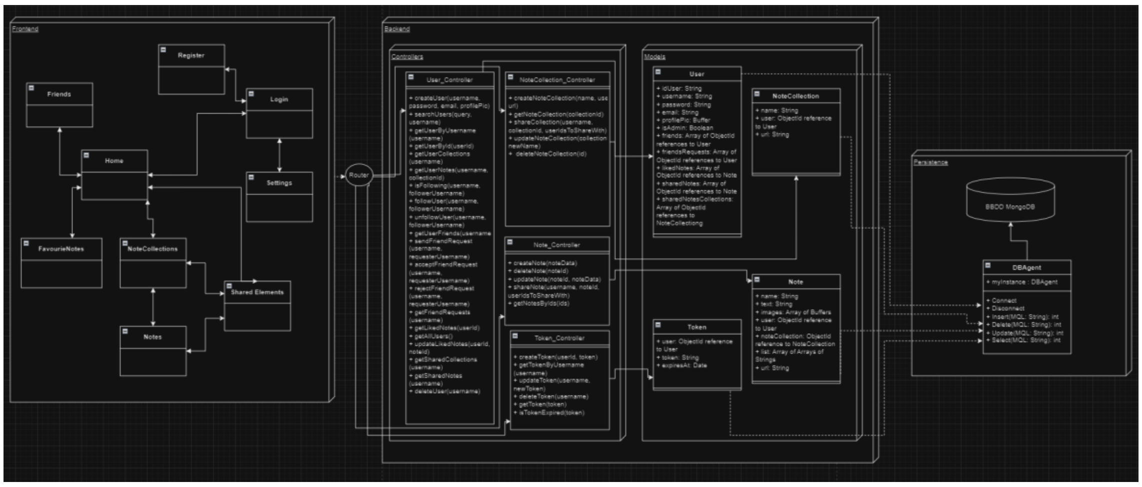
2.2. Navigation Diagram.

In the following image we can appreciate how the navigation inside the Web App will be managed, highlighting those functionalities that stand out:



2.3. Presentation Diagram.

Here we can see the resultant content Diagram for the webApp:



3. User Manual.

In this section we are going to discuss what technologies and how the webApp must be executed to work as expected. Firstly, we must have Linux/Ubuntu as OS.

3.1. Installation Guide.

First, we must install **NodeJS** and **Ruby on Rails** using the following pdf as a guide:

- RoRNJS - Installation guide.pdf

Now, all we must do is to install the dependencies of our project (these dependencies can be found in package.json for both backend and frontend) and then execute all files that are on the GitHub Repository (see last page of the document to get the link).

First, we must input this command on the **backend**:

- *npm install*
- *npm start*

Then, we execute the **frontend**, using the same commands:

- *npm install*
- *npm start*

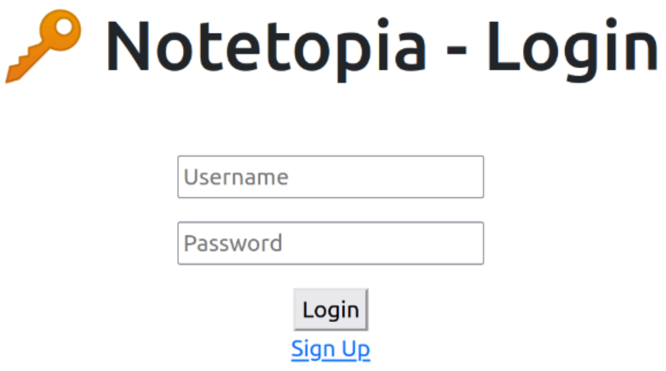
Now to get the page up and running we must use our browser and search for localhost on port 3001, like this:

- *localhost:3001*

3.2. User Guide.

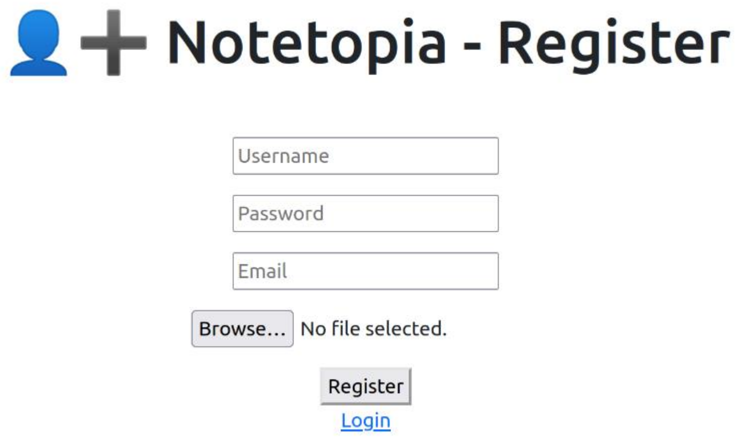
3.2.1. Login Page

What we find here is two inputs, one for the username and another for the password. We also found 2 buttons, one for login the user into the application and another for the register page:



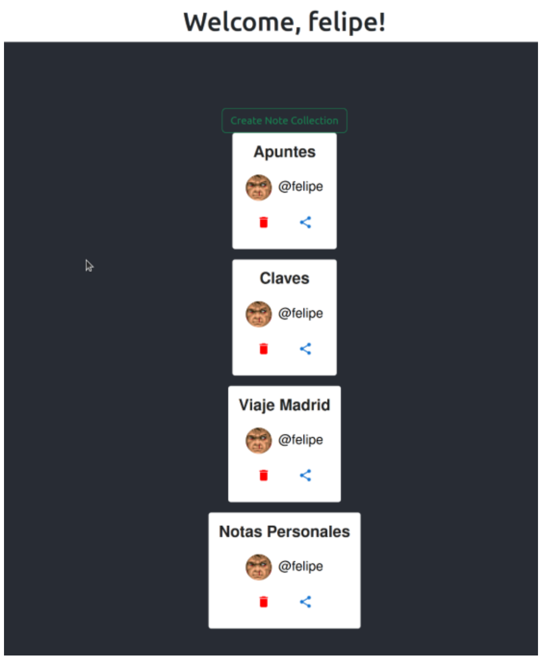
3.2.2. Register Page

Here we find a page with 4 inputs. One for the username, another for the password, another for the email and finally, another for the profile picture. The profile picture has a button that if you press it, you can choose a photo from your computer. Also, we find 2 bottoms, one to register the new user and other button to go back to the login page:

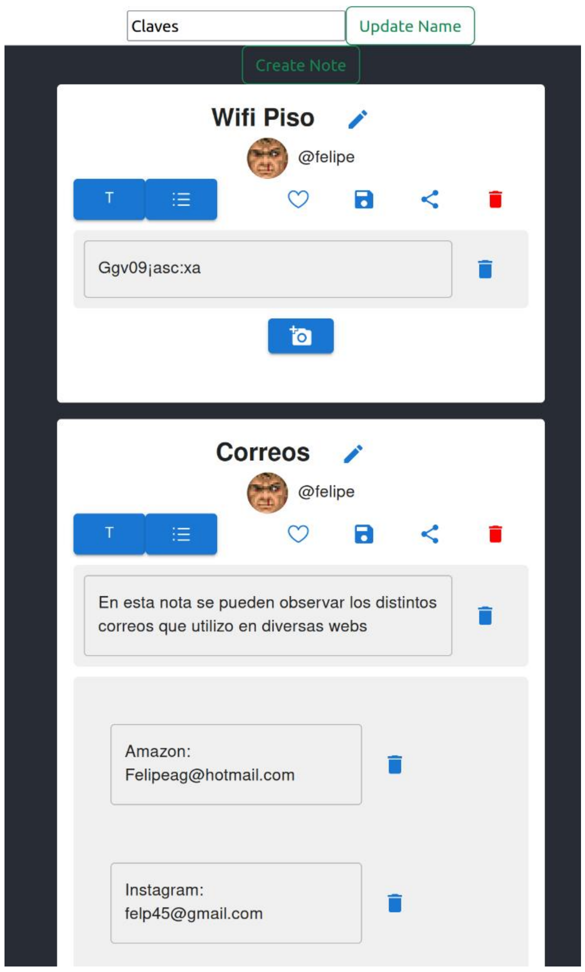


3.2.3. Home

The home page is the main page we see when we perform the login. Here our note collections will show up, also allowing us to create new ones, sharing them or delete them:



After clicking inside a note collection, you can see all notes inside and add new ones. Inside a note, you can like it, share, delete and there's a button to save the changes performed to the note. On the upper side of the page, we can change the note Collection Name and above the user we can modify that specific note name:



Ggv09¡asc:xa

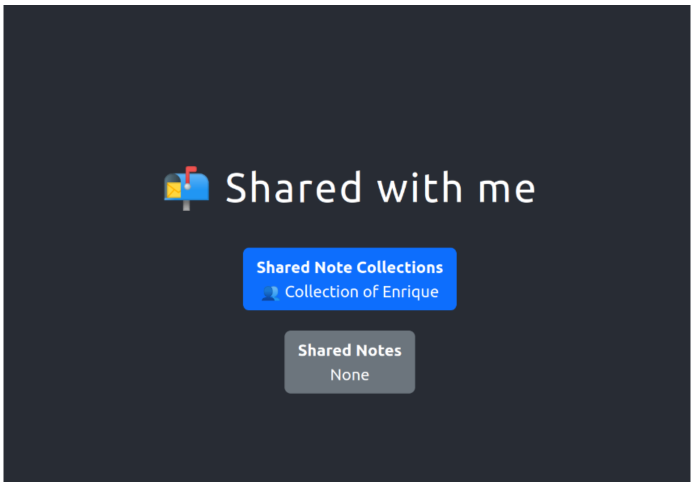
En esta nota se pueden observar los distintos correos que utilizo en diversas webs

Amazon:
Felipeag@hotmail.com

Instagram:
felp45@gmail.com

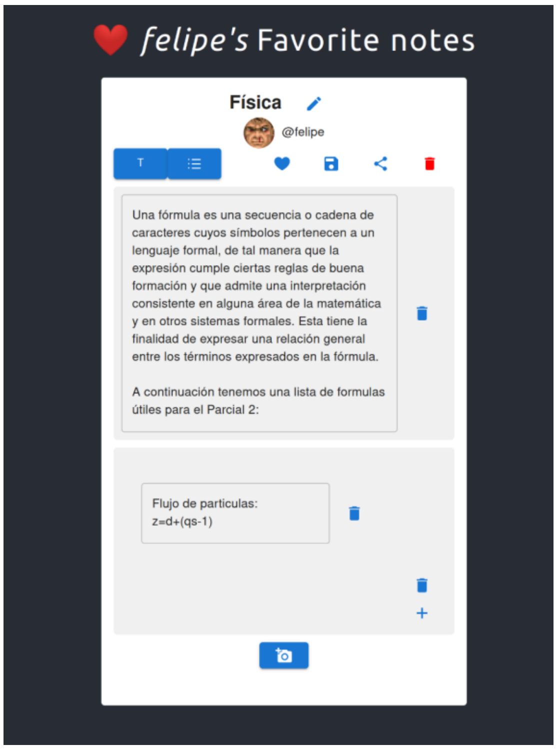
3.2.4. Shared with me

In this page we can see the shared notes and shared note collections from friends. These notes can only contain those that a friend has shared with you, and they behave just as if they were yours. You can see a small preview with their names:



3.2.5. Favourite Notes

In this page we can see the liked notes. These notes can be editable the same way if you have access them with their collectionNote:



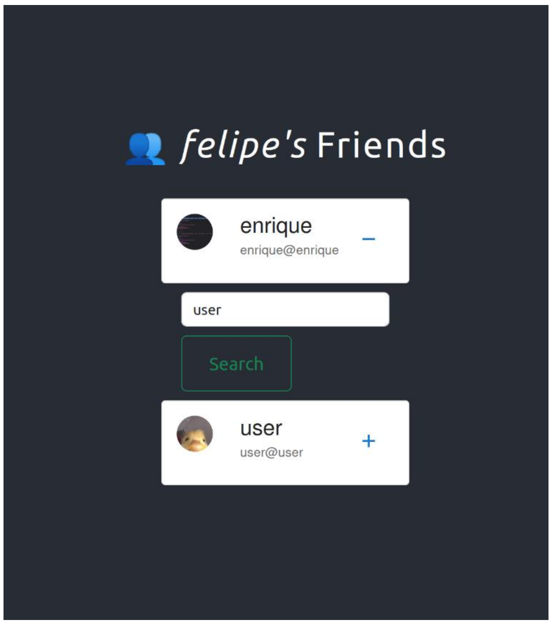
Una fórmula es una secuencia o cadena de caracteres cuyos símbolos pertenecen a un lenguaje formal, de tal manera que la expresión cumple ciertas reglas de buena formación y que admite una interpretación consistente en alguna área de la matemática y en otros sistemas formales. Esta tiene la finalidad de expresar una relación general entre los términos expresados en la fórmula.

A continuación tenemos una lista de formulas útiles para el Parcial 2:

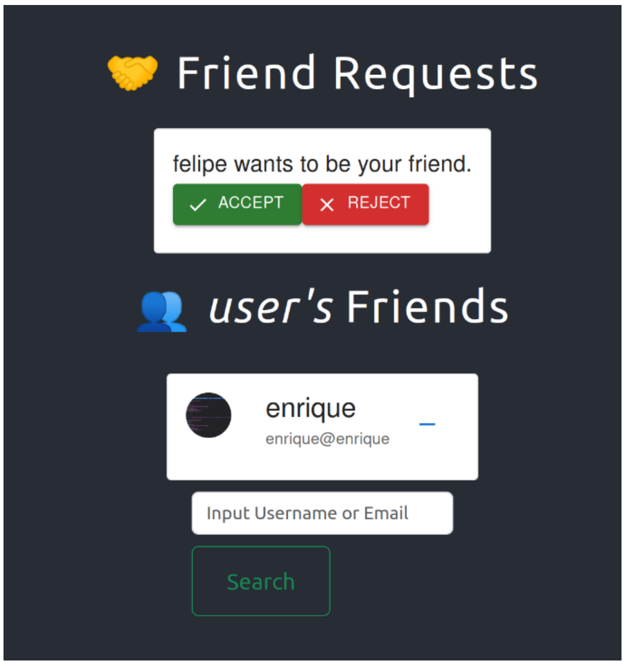
Flujo de particulas:
z=d+(qs-1)

3.2.6. Friends

Here we can see the list of friends as well as a search bar to send friend requests to other users from the app:



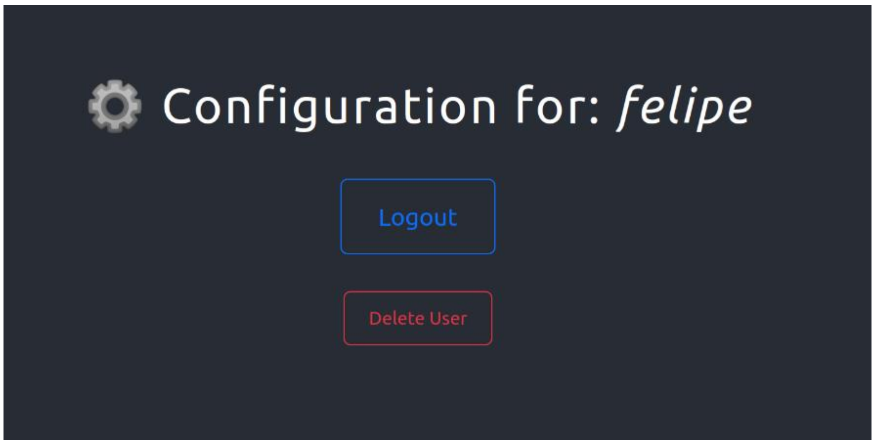
Moreover, friend requests can be accepted or rejected depending on the user's choice. A pop up will show up whenever a friend request is sent:



3.2.7. Configuration

Here we can see the configuration page where you can logout and sigh in with another user or delete the actual logged user.

In this configuration page we can see two options for the user. One to simply logout and another one to permanently delete the user:

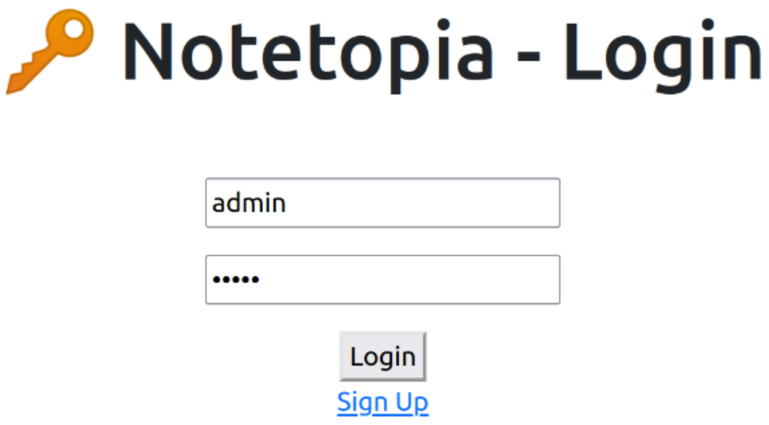


3.2.8. Administrator

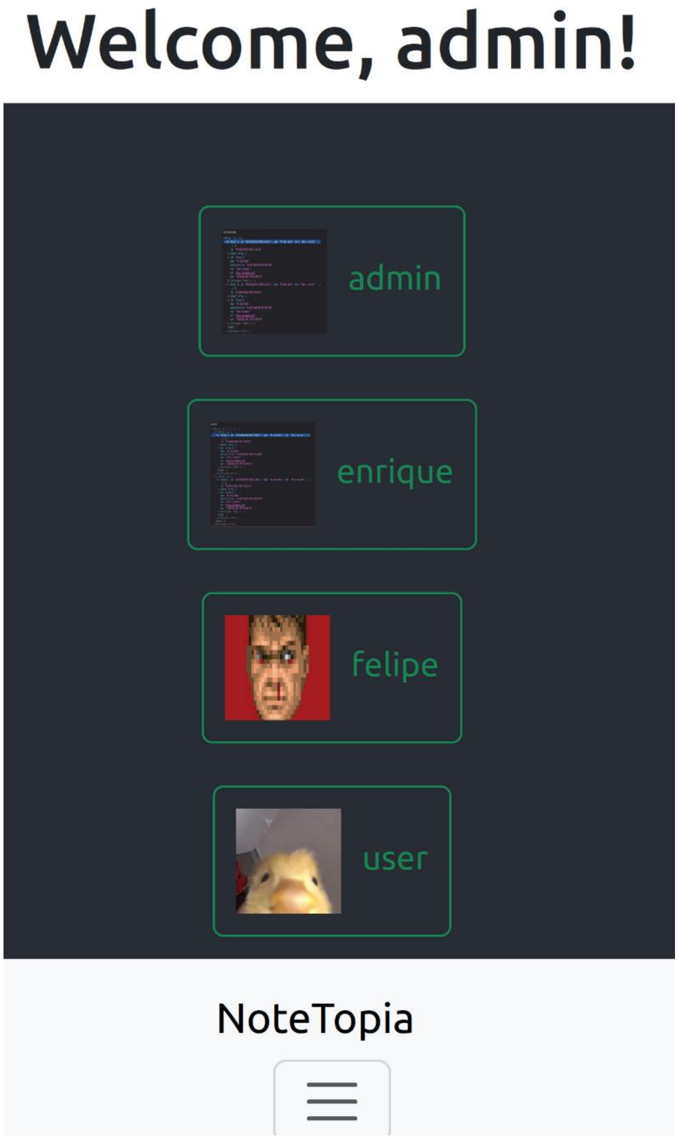
Also, we have created a administrator user. When we login with this user, we will see a new page with all the users:

User: admin

Password: admin



In that page we can login in the application as any user and do any configuration we want as shown here:



4. API Dictionary.

As we have seen in the Diagrams, we have created four files that contains the methods that are in our API Dictionary.

- Index.js

Verb	***Path***	***Action***	***Used for***
POST	/	render	Loads all the windows

- login.js

Verb	***Path***	***Action***	***Used for***
POST	/login	login	Authenticate a user

- register.js

Verb	***Path***	***Action***	***Used for***
POST	/register	Register	Register a user

- users.js

Verb	***Path***	***Action***	***Used for***
GET	/:username/collections	show	Show user collections
GET	/:username/notes	show	Show user notes
DELETE	/:username	destroy	Delete an user
DELETE	/collections/:collectionId	destroy	Delete a noteCollection
POST	/:username/collections	create	Create a new note collection
POST	/:username/notes	create	Create a new note
PUT	/noteCollectionsUpdate/:collectionId	update	Update a note collection
GET	/noteCollections/:collectionId	show	Show a specific note collection
GET	/search	show	Search for users
POST	/:username/follow	create	Follow a user
POST	/:username/unfollow	create	Unfollow a user
GET	/:username/is-following/:followerUsername	show	Check if a user is following another user
GET	/:username/friends	show	Get user's friends
POST	/:username/friend-request	create	Send a friend request
POST	/:username/accept-friend-request	create	Accept a friend request
POST	/:username/reject-friend-request	create	Reject a friend request

GET	/:username/friend-requests	show	Get user's friend requests
GET	/:username/:collectionId/notes	show	Show notes from a specific collection
DELETE	/notes/:noteId	destroy	Delete a note
PUT	/:userId/likedNotes	update	Update liked notes for a user
GET	/:userId/getLikedNotes	show	Get liked notes for a user
POST	/:username/share-collection	create	Share a note collection
POST	/:username/share-note	create	Share a note
GET	/:username/sharedNoteCollections	show	Get shared note collections
GET	/:username/sharedNotes	show	Get shared notes
PUT	/notes/save/:noteId	update	Save a note
GET	/getAllUsers	show	Get all users
GET	/:collectionId/getUserUsername	show	Get username from a note collection
GET	/:username/collections	show	Show user collections
GET	/:username/notes	show	Show user notes
DELETE	/:username	destroy	Delete an user

5. GitHub Link.

The link to the Github Repository is the following, here we can see all the code that has been developed along with every single commit that was performed on the development of the app:

https://github.com/Enriquesmo/DSW-NoteTopia.git