

Report



Context and goal	2
Technologies	2
Web application	3
Homepage	3
Offer page	4
Register and login page	5
Details page	6
Managing offers page	7
Database	8
Portability	9
Difficulties	9
Possible improvement	10

Supervisors : Didier Donsez

1. Context and goal

This project is a development of a web application. It has been created for an association, "Des'Alter cafés".

The associative life of this association is to create and animate optimistic cafés where it is good to work, have lunch, help each other in a zero waste logic. The association contributes to the choice of the program, and participates in the animation of the evenings/workshops. It helps social entrepreneurs or project leaders by:

Information / advertising (newspaper).

Networking with a wider audience.

The promotion of a spirit of mutual aid through fun and friendly tools and activities

The spirit of mutual aid between and around social entrepreneurs, will be honored during monthly evenings "barter talents / Social barter" where people can offer help, find interested people. They will be invited to thank those who help them by offering them a drink(s) in the café.

The purpose of this project is to allow social entrepreneurs to contact each others and share their skills via the web application. They will come on the web application and will be able to see offers from others, notify their interest, ask for an appointment or create an offer themselves.

2. Technologies

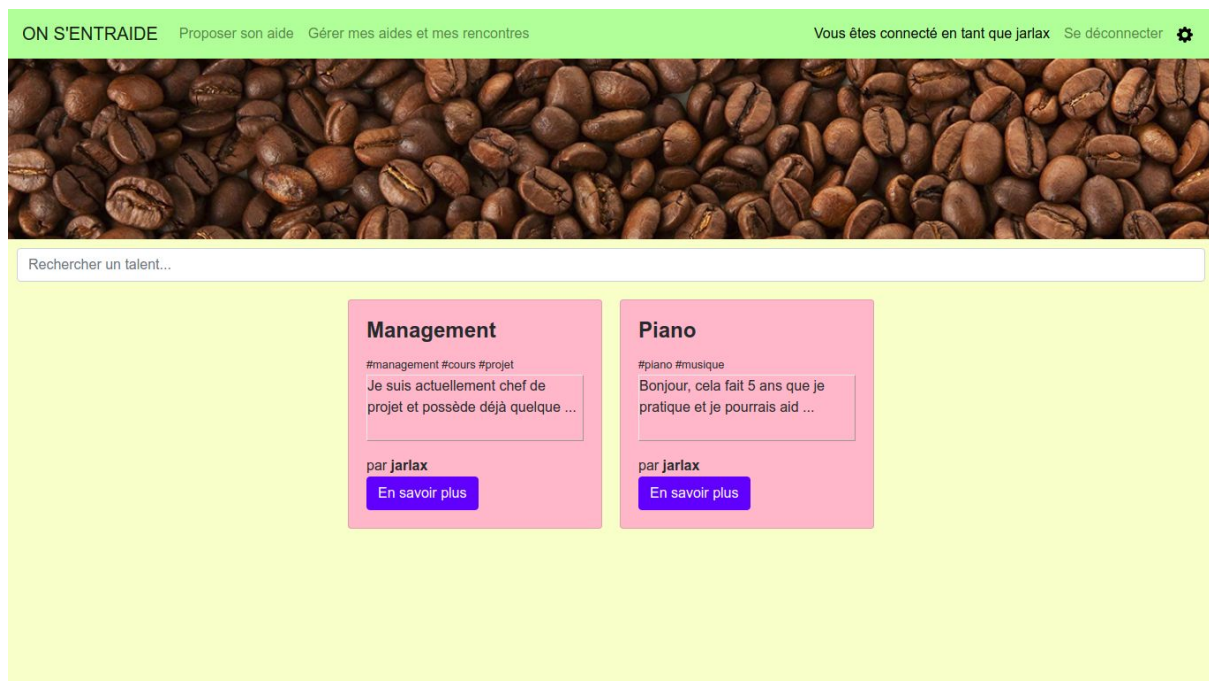
Meteor was our framework for this project which allow us to create a web application. For the database (where all the help are are listed, with the name, the description, the schedule, etc.) we used MongoDB which is linked and included in meteor.

For the style of the application, we used the Bootstrap framework.

For mail sending, we used mailgun which allow us to send mail confirmation when a user create an account on our website.

3. Web application

a. Homepage



The home page is where the various offers can be found. Using a search bar, offers can be sort by the words typed in the search bar. This will display the offers containing the words written in it in their title, description or keywords. Our client wanted the offers display in the style of post it notes.

At the top, there is the navigation bar which is used to navigate between pages. The button "Proposer son aide" send the user to the page to create an help offer. The button "Gérer mes aides et mes rencontres" send the user on the page where he can see the offers he created. The button to log in or log out can be found on this bar as well as the parameters button.

Back end regarding, it will just display all the database where you can find all the offers, and display what he have to be display (our client wanted the name of the offer, the pseudo of the user who propose it, part or the description, and keywords). The search bar send queries to the database of help offers and then display the result. The user can search any words, it will try to find it in the title, description, keywords). We used regular expression, a user can search a part of a word for example in the searchbar and it will try to find something which looks like this word. For instance, if we search RI, it will return a title where RICM appear.



b. Offer page

The image shows a screenshot of a form titled 'Descriptif de l'offre' (Offer Description) set against a background of coffee beans. The form is on a light yellow background and contains the following fields and elements:

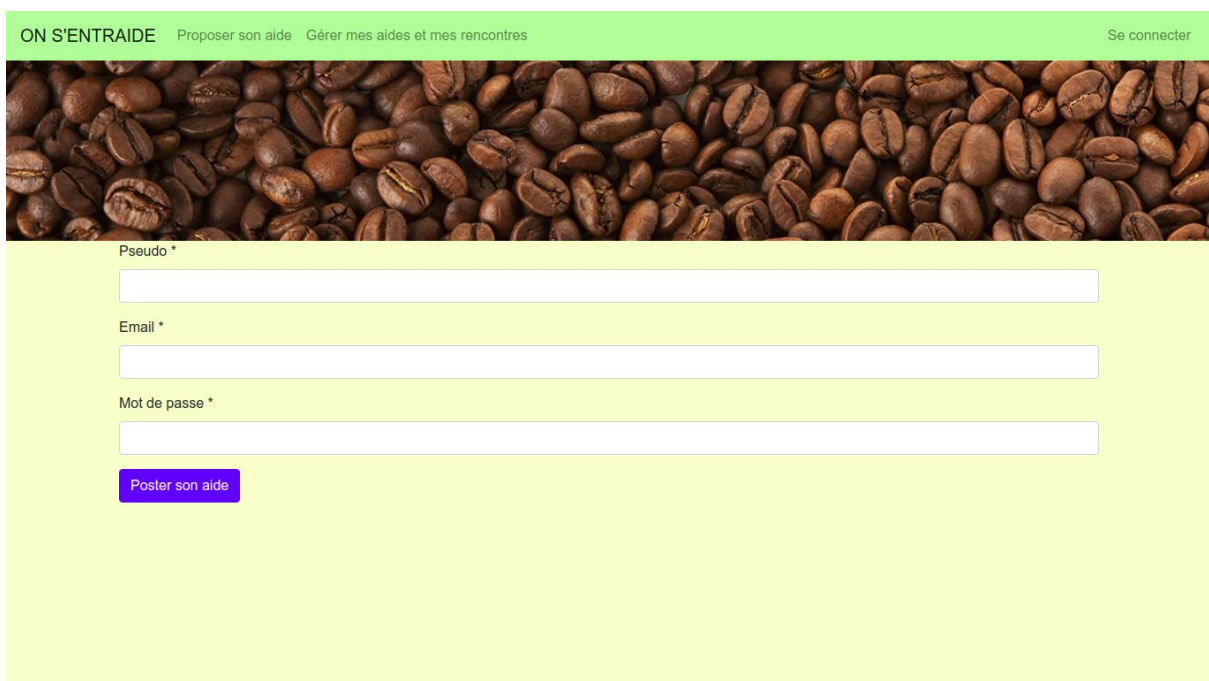
- Titre** (Title): A text input field with the placeholder 'Le titre de votre aide'. Below it, a note says 'Votre titre doit avoir moins de 30 caractères.'
- Description**: A large text area with the placeholder 'Description'. Below it, a note says 'Votre description doit avoir moins de 100 caractères.'
- Choix de mots-clés** (Choice of keywords): A text input field with the placeholder 'Mots-clés'. Below it, a note says 'Liste de mots-clés séparés par des espaces.'
- Vous êtes disponible :** (You are available): Two blue buttons: 'Une ou plusieurs fois par semaine' and 'Une date précise'.
- Dans quel café ?** (In which cafe?): A dropdown menu with 'Des'Alter Cafe' selected.
- Poster mon aide**: A red button at the bottom.

The offer page is where someone can add a new helps. If someone have a skill and want to share it, he just have to go on this page, and then, complete all the blanks below. It contain the name this help, a description, when he's available, and the name of the café he want to do the meeting. For the moment, there is only one café, Grenoble's one, but it could be change if there is another café in the future. After entering all those data, he just have to click on the submit button. After that there is two cases, if he's connected, the offer is directly added to the homepage. But if he's not connected, it will be asked to register or log in, so, he will be redirected to register and log in page. At the moment he will be connected or register, his offer will be added.


If he's not connected, we chose to use a session variable to keep all information about the offer waiting the moment he will be registered.

If he's connected, we add all the data directly in the database where we can find all the offers .

c. Register and login page



ON S'ENTRAIDE [Proposer son aide](#) [Gérer mes aides et mes rencontres](#) [Se connecter](#)



Pseudo *

Email *

Mot de passe *

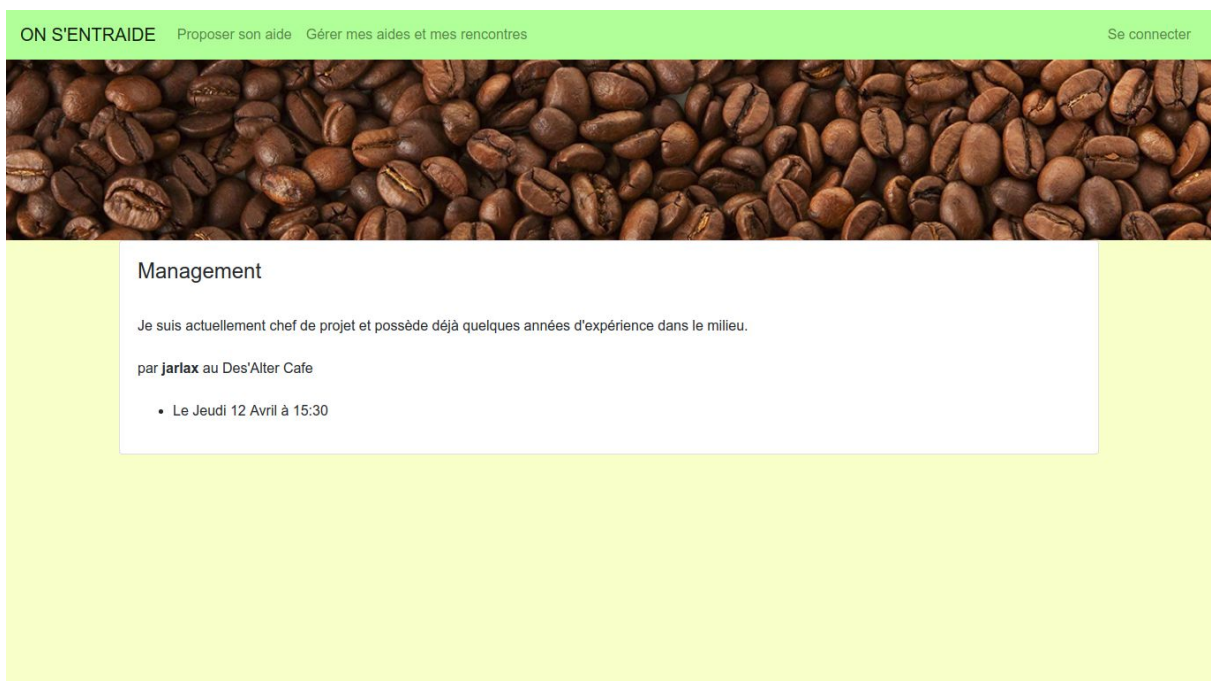
[Poster son aide](#)

Our client wanted to minimize the log in/sign up system so that the users won't feel compelled to register in the application to browse the offers, and so she wanted a page which would do both, register and sign in but which will only appear at the end of the process of creating an offer or asking for an appointment.

This form will check if so username and the email given are already used, if they are, it will then check if the email matches the username in the database and if it does, it will try to connect the user using the password. If the username and the email doesn't match an error message will be prompted.

If the username and the email are not used, an account will be created and connected. A confirmation email will be sent, and will need to be confirmed before the offer is displayed.

d. Details page



The screenshot shows a web application interface. At the top, there is a green navigation bar with the text "ON S'ENTRAIDE" and two links: "Proposer son aide" and "Gérer mes aides et mes rencontres". On the right side of the bar is a link "Se connecter". Below the navigation bar is a large image of coffee beans. Underneath the image is a white box containing the following text:

Management

Je suis actuellement chef de projet et possède déjà quelques années d'expérience dans le milieu.

par **jarlax** au Des'Alter Cafe

- Le Jeudi 12 Avril à 15:30

On this page, the user can see more about the offer he clicked on the homepage. The user can see here, the time windows and the café where the appointment is expected.

e. Managing offers page

ON S'ENTRAIDE Proposer son aide Gérer mes aides et mes rencontres Vous êtes connecté en tant que jarlax Se déconnecter ⚙️

Mes aides proposées

Management • Le Jeudi 12 Avril à 15:30

Piano • Les Lundi entre 12:30 et 12:30 • Les Mardi entre 15:30 et 15:30

This page allow a user to manage his offers. For example, if the user did a mistake in the name of the offer or in the schedule, he will be able to modify it. Also, he can remove the offer from the website if he wants, just clicking on the garbage icon.

To modify an offer, the user just have to click on the pen. Then, it will bring him on the offer page we saw above. It look likes the same, but it's not the same page. On this one, he will have all the information he had when had created the offer. He just have to modify what he want to modify, and then, there is the same button at the bottom of the page which can allow him to validate his modification. At the end of this operation, the offer is modified in the database and so everywhere on the website.

Back end regarding, we use session variable to keep information when we change on the other page. For the deletion, it will just delete the offer from the database directly.

4.Database

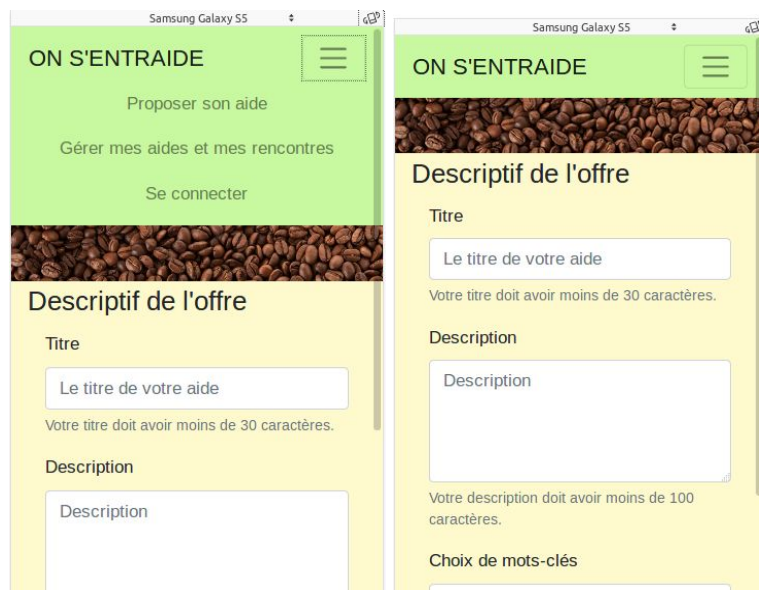
For the database, we used mongoDB which is linked with meteor. We have only one class which is all the helps provided by all the users. For one help, we have a unique `_id` of the offer which is the key of this class. We have a Title, a Description of the help, the Café where the meeting should take place, some Keywords about the offer. Then, there is a `preciseDate` et `Precisehour`. This values are filled if the user chose to provide his help only one time, it's "null" otherwise. Same thing for `tabCreneau`, `tabStartHour` and `tabEndhour`. It's an array of days for the first one ('Lundi','Mardi',...) and the 2 others one are hours (like 12:12pm). We check the first element of those 3 array for exemple to see when is the first schedule. We print this in the details page if someone is interested about this offer and he wants to know when it's going to be. There is the `userId` which is an id again but for the user. The last field is the Status. If the user create his offer but he's not connected yet or he didn't have an account, we add it to the database but with "Attente" which mean that the offer his not going to be show on the home page. When the user is connected, the status is "Envoyé" which mean that the Help can be show.

It's the only class we created but, with meteor, there is another class in the database. It's not handle by us. When you create an account on this website, it's automatically added in this class.

Help
<code><u>id</u></code>
<code>title</code>
<code>Description</code>
<code>Café</code>
<code>CreatedAt</code>
<code>Keywords</code>
<code>Percisedate</code>
<code>Precisehour</code>
<code>tabCreneau</code>
<code>tabStartHour</code>
<code>tabEndHour</code>
<code>userId</code>
<code>Status</code>

5. Portability

Our client asked us to have a website which can be very accessible for smartphone. So you can see below how it's look like on a smartphone screen. Bootstrap is used for that because of its responsive classes which works whether you are on a computer or on a mobile.



6. Difficulties

The most difficult part was to learn how to use the Meteor framework and its packages. Like how to use the spacebars or the templates.

Also, learning how to code in JavaScript was a little difficult since we never worked on a project using it.

Sending confirmation email was easy because it is included in the account package but it was a bit difficult to set up the sending email. So we use mailgun to do it. It comes with a test email we can use to send email to only email addresses that are authorized. But can also use your domain email if you have a domain name on internet.

For page change, we started by using dynamic templates because we thought that a router was difficult to set up. Finally we decided to try it, using iron router, and it was easy, allowing us to do many things beside just changing pages.

7. Possible improvement

This project is over for what we had to do, but he's not over. We developed all the features that were requested for the first version, but with the idea that someone could take over the application to continue to develop it after.

We had two main things that we haven't the time to do.

First, handling the appointments. The application should have the ability to handle appointments and sending email if someone is interested, and then, the one who give the offer, should have been able to answer back if he's available or not.

Also, the security side. The web application is not ready to be deployed online right now because the database is not secure and every client can modify the database like they want. We need to modify the system using publish/subscribe system of meteor.

We didn't have the time to change color of the post it in the first page, we could had a function which give random color (but pre-selected) to give more color to the site (they are all pink for the moment).

Finally, we didn't have the time to deploy the application on a website, but it is the last step before everyone could be able to use it.