RAPPORT
MINI PROJET SYMFONY
GESTIONNAIRE DE CONTACTS

Papa Oumar DIEYE

ESMT DAKAR

Sommaire

I. Fonctionnement de l'application

A. Ajouter un contact
B. Afficher la liste des contacts
C. Afficher les coordonnées d'un contact
D. Modifier les coordonnées d'un contact
E. Supprimer un contact

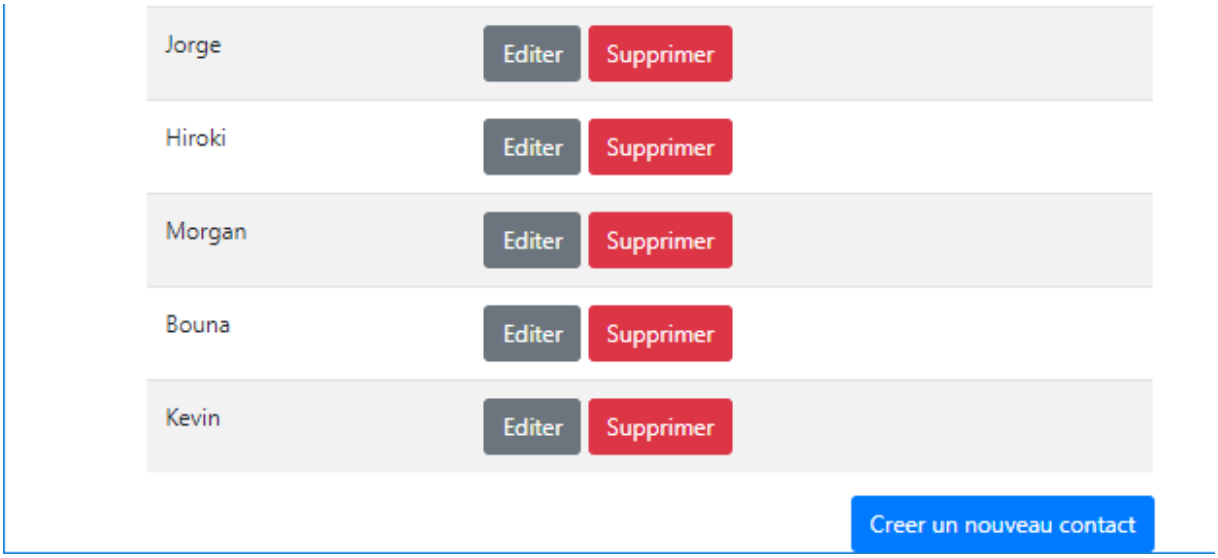
II. Ajout apporté à l'application

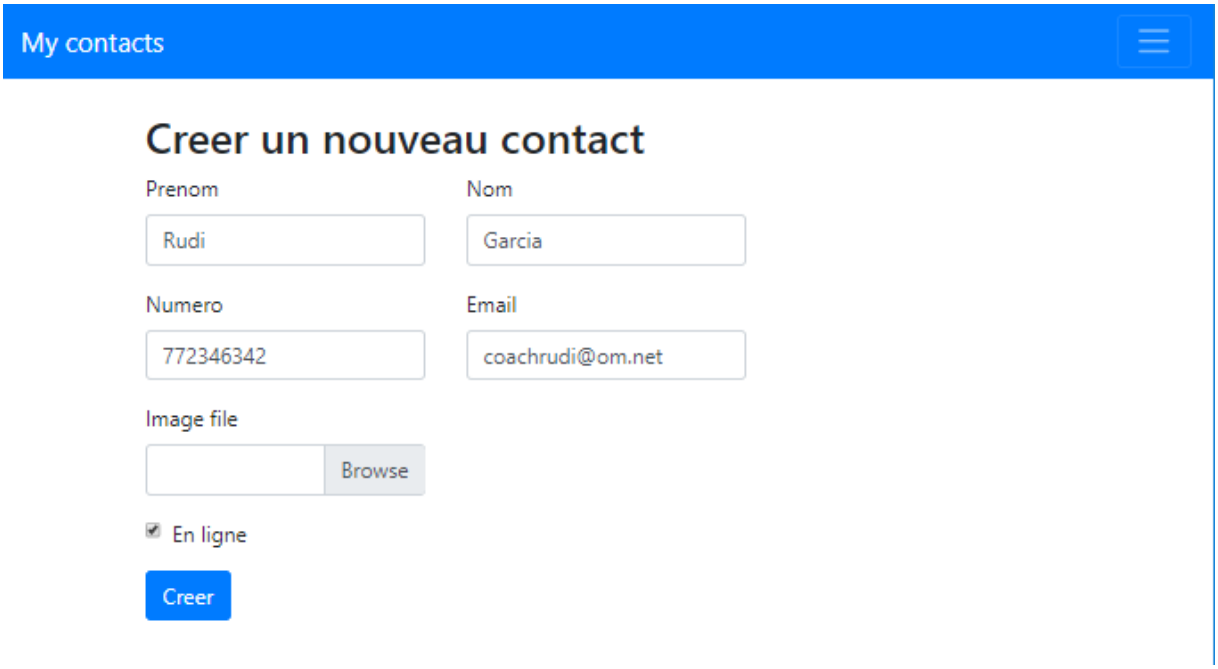
A. Authentification
B. Pagination
C. Image a la une
D. Filtrage des contacts

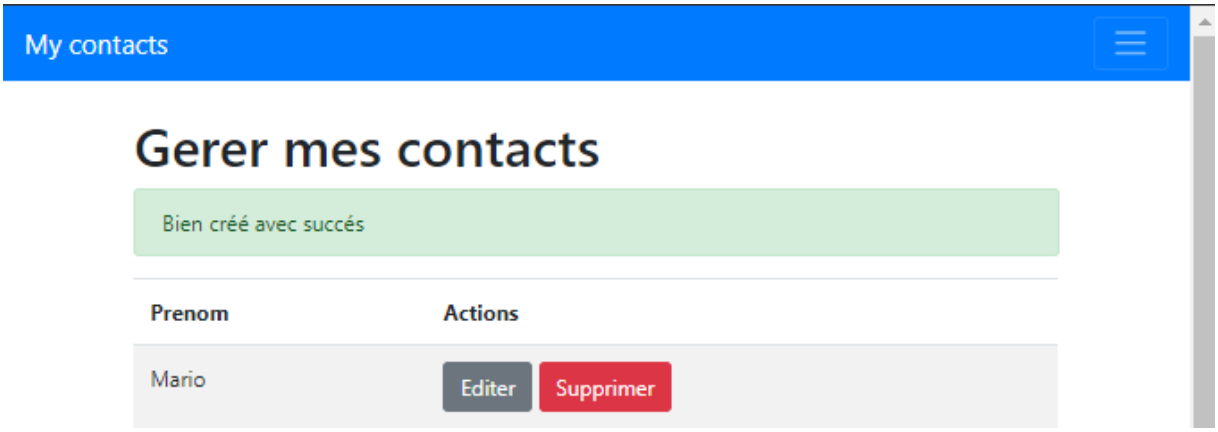
I. Fonctionnement de l'application

A. Ajouter un contact

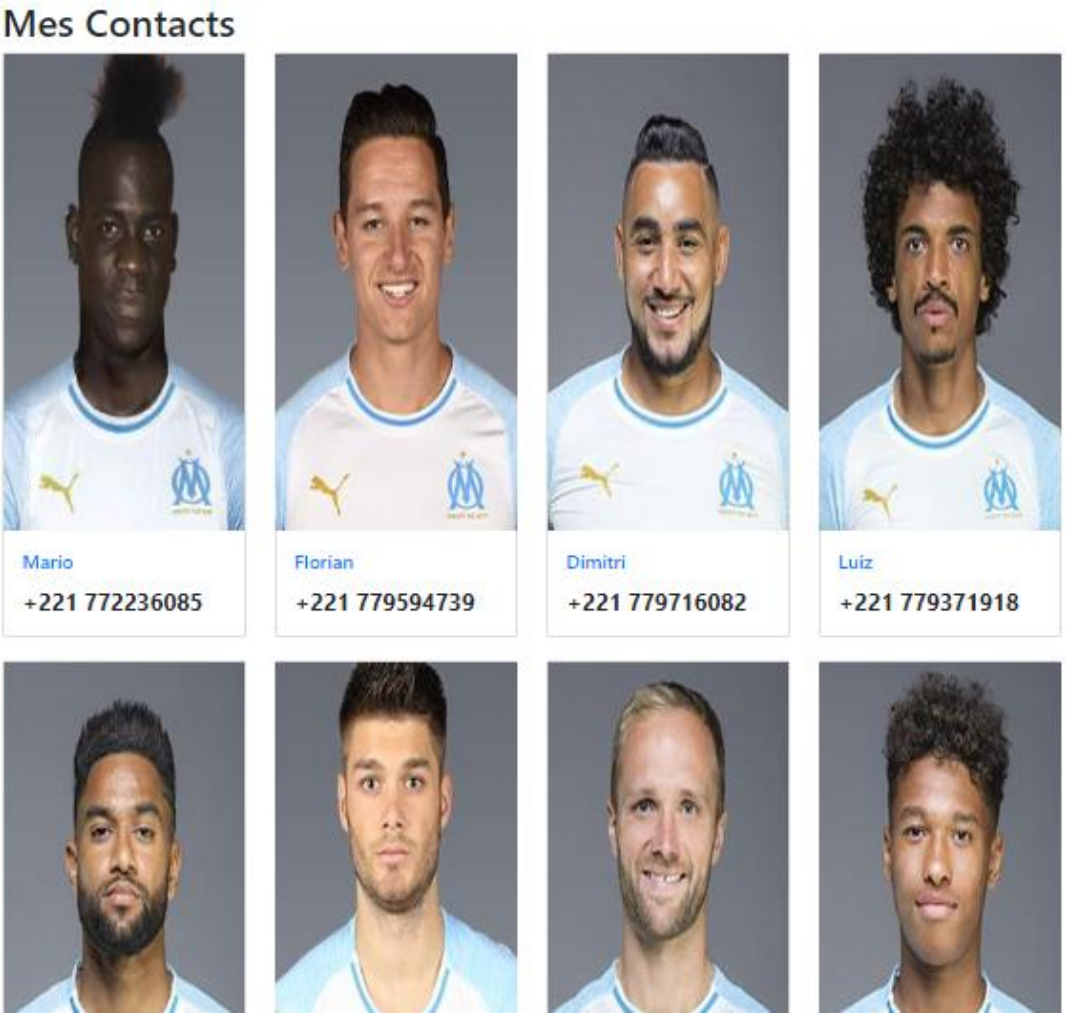
```php
/**
 * @Route("/admin/contact/create", name="admin.contact.new")
 * @param Request $request
 * @return RedirectResponse|Response
 * @throws \Exception
 */

public function new(Request $request) {
    $contact= new Contact();
    $form = $this->createForm( type: ContactType::class, $contact);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()){
        $this->em->persist($contact);
        $this->em->flush();
        $this->addFlash( type: 'success', message: 'Bien créé avec succés');
        return $this->redirectToRoute( route: '/admin');
    }
    return $this->render( view: 'admin/contact/new',[
        'contact' =>$contact,
        'form'=>$form->createView()
    ]);

}
```




My contacts
Creer un nouveau contact
Prenom
Rudi
Nom
Garcia
Numero
772346342
Email
coachrudi@om.net
Image file
Browse
En ligne
Creer

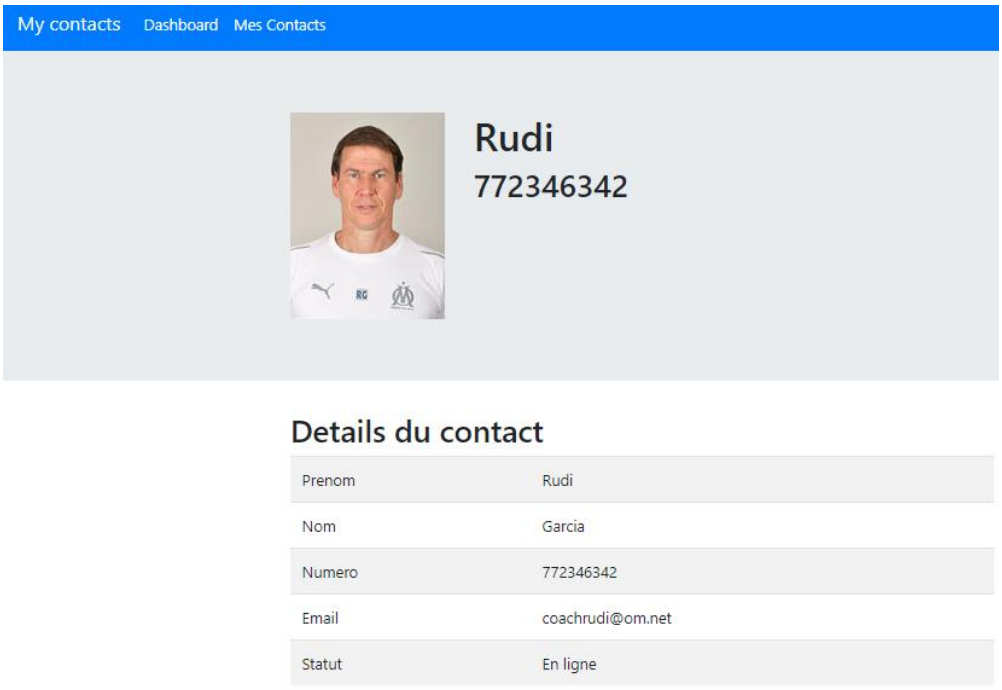

My contacts
Gerer mes contacts
Bien créé avec succés
Prenom
Actions
Mario
Editer
Supprimer

B. Afficher la liste des contacts




```
<h1>Mes Contacts</h1>
<div class="row">


    {% for contact in contacts %}
        <div class="col-md-3">
            {% include 'contact/_contact' %}
        </div>
    {% endfor %}

</div>
```

C. Afficher les coordonnées d'un contact




```
<div class="row">
    <div class="col-md-8">


        <h2>Details du contact</h2>
        <table class="table table-striped">
            <tr>
                <td>Prenom</td>
                <td>{{ contact.prenom }}</td>
            </tr>
            <tr>
                <td>Nom</td>
                <td>{{ contact.nom }}</td>
            </tr>
            <tr>
                <td>Numero</td>
                <td>{{ contact.numero }}</td>
            </tr>
            <tr>
                <td>Email</td>
                <td>{{ contact.email }}</td>
            </tr>
            <tr>
                <td>Statut</td>
                <td>{{ contact.statusType }}</td>
            </tr>

        </table>
    </div>
    <div class="col-md-4">

    </div>
```

D. Modifier les coordonnées d'un contact

```php
/**
 * @Route("/admin/contact/{id}", name="admin.contact.edit", methods="GET|POST")
 * @param Contact $contact
 * @param Request $request
 * @param CacheManager $cacheManager
 * @param UploaderHelper $helper
 * @return \Symfony\Component\HttpFoundation\Response
 */
public function edit(Contact $contact, Request $request, CacheManager $cacheManager, UploaderHelper $helper){

    $form = $this->createForm( type: ContactType::class, $contact);
    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()){
        if ($contact->getImageFile() instanceof UploadedFile) {
            $cacheManager->remove($helper->asset($contact, fieldName: 'imageFile'));
        }
        $this->em->flush();
        $this->addFlash( type: 'success', message: 'Bien modifié avec succés');
        return $this->redirectToRoute( route: '/admin');
    }

    return $this->render( view: 'admin/contact/edit',[
        'contact' =>$contact,
        'form'=>$form->createView()
    ]);

}
```

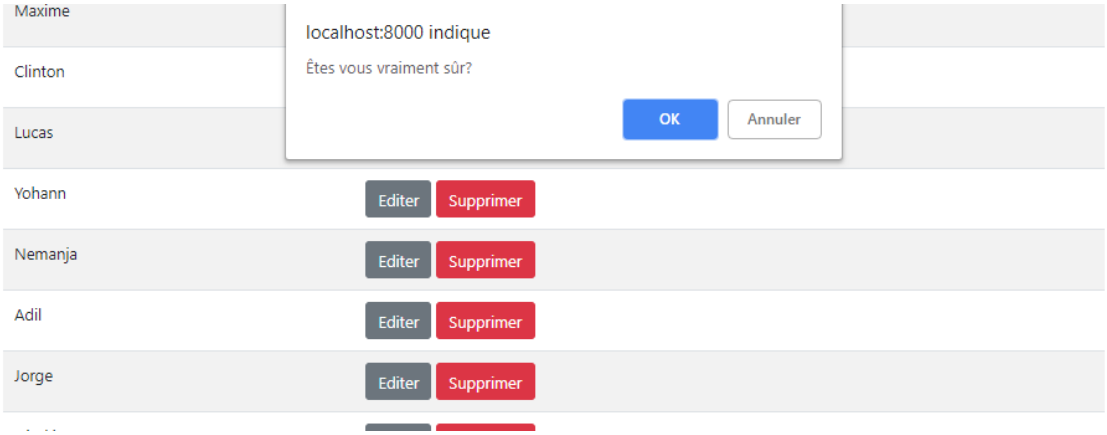
Gerer mes contacts

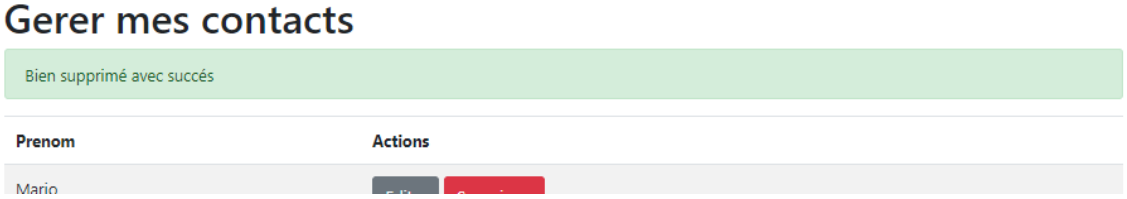
Bien modifié avec succés

Prenom	Actions
Mario	Editer Supprimer

E. Supprimer un contact

```php
/**
 * @Route("/admin/contact/{id}", name="admin.contact.delete", methods="DELETE")
 * @param Contact $contact
 * @param Request $request
 * @return RedirectResponse
 */
public function delete(Contact $contact, Request $request){

    if ($this->isCsrfTokenValid( id: 'delete'. $contact->getId(), $request->get( key: '_token'))){
        $this->em->remove($contact);
        $this->em->flush();
        $this->addFlash( type: 'success',  message: 'Bien supprimé avec succès');
    }

    return $this->redirectToRoute( route: '/admin');
}
```





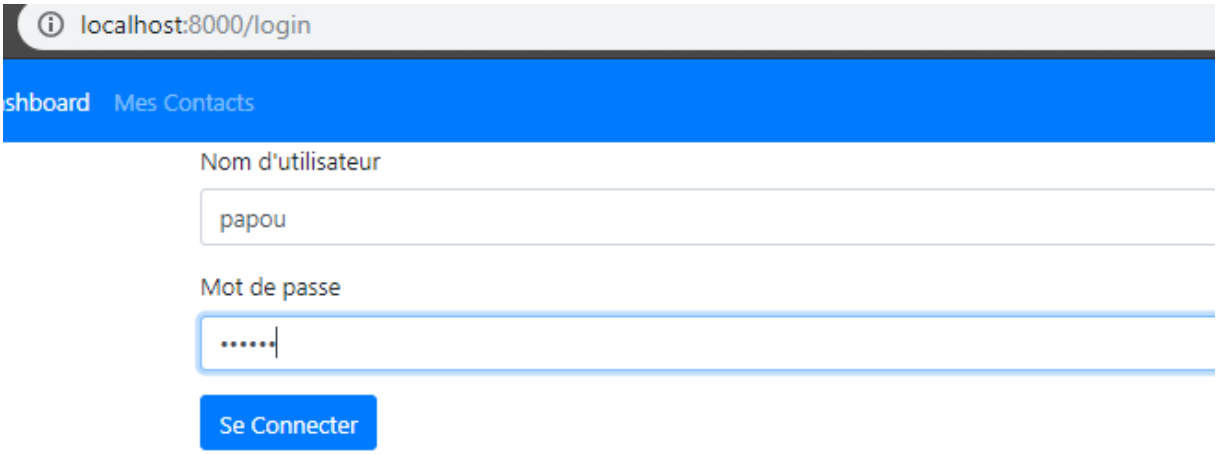
II. Ajout apporté à l'application

A. Authentification

Pour l'authentification on a mis en place un système qui permet seulement à l'administrateur de pouvoir effectuer les opérations telles que l'édition, la création et la suppression de contact.

```php
class SecurityController extends AbstractController{

    /**
     * @Route("/login", name="login")
     * @param AuthenticationUtils $authenticationUtils
     * @return \Symfony\Component\HttpFoundation\Response
     */

    public function login(AuthenticationUtils $authenticationUtils )
    {
        $error = $authenticationUtils->getLastAuthenticationError();
        $lastUsername = $authenticationUtils->getLastUsername();
        return $this->render( view: 'security/login',[
            'last_username'=>$lastUsername,
            'error'=>$error
        ]);
    }
}
```

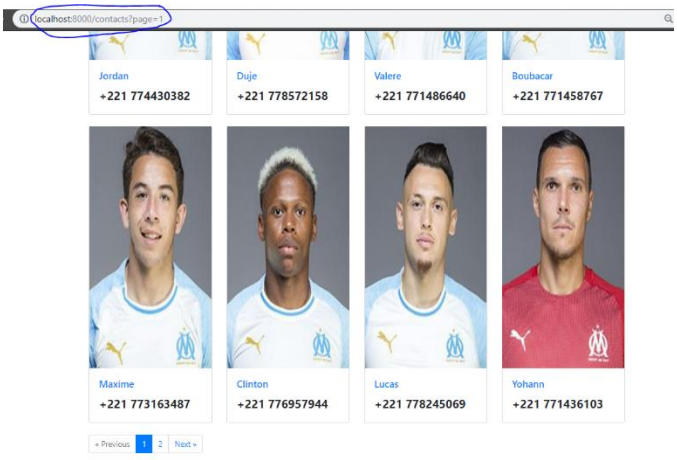
```twig
{% extends 'base' %}

{% block title 'Se Connecter' %}
{% block body %}
    <div class="container">
        {% if error %}
            <div class="alert alert-danger">
                {{ error.messageKey |  trans(error.messagedata, 'security')}}
            </div>
        {% endif %}
        <form action="{{ path ('login') }}" method="post">
            <div class="form-group">
                <label for="username">Nom d'utilisateur</label>
                <input type="text" id="username" name="_username" class="form-control" value="{{ last_username }}">
            </div>

            <div class="form-group">
                <label for="password">Mot de passe</label>
                <input type="password" id="password" name="_password" class="form-control">
            </div>
            <button type="submit" class="btn btn-primary">Se Connecter</button>
        </form>
    </div>
{% endblock %}
```

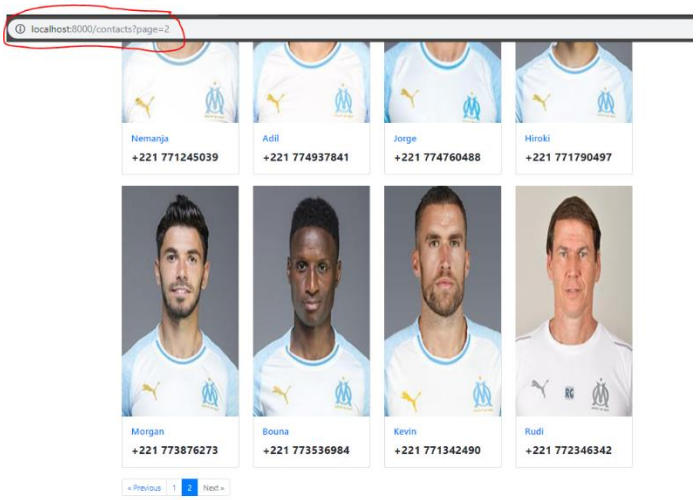

localhost:8000/login
shboard
Mes Contacts
Nom d'utilisateur
papou
Mot de passe
Se Connecter

B. Pagination

Ici on a listé nos contacts avec une pagination dans notre cas c'est 10 contacts par pages.

On a du télécharger un bundle pour la pagination qui prend en paramètre le référencement(KnpPaginatorBundle)

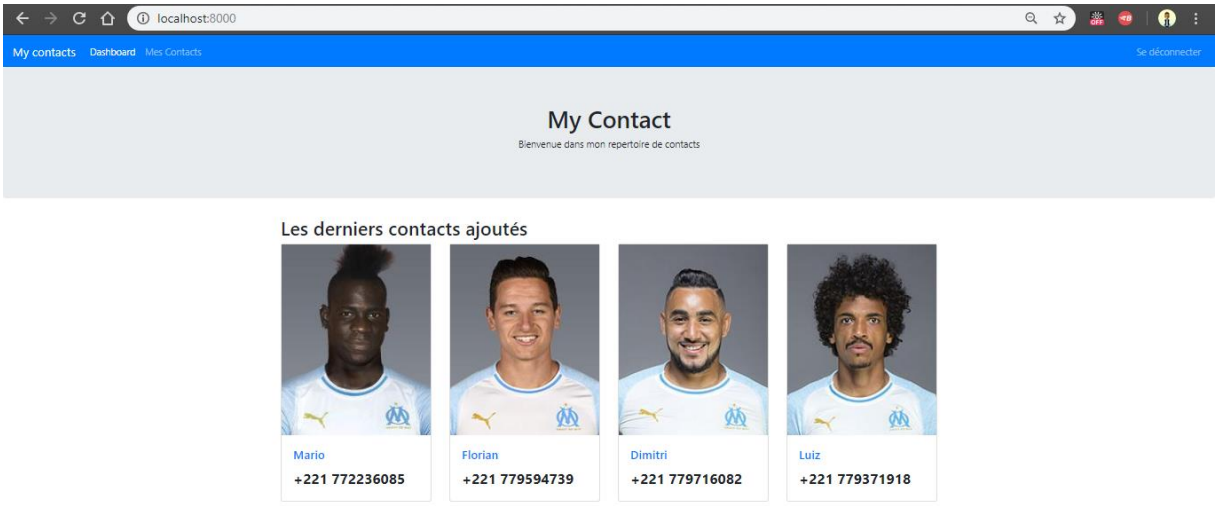





```
<div class="navigation">
    {{ knp_pagination_render(contacts) }}
</div>

/div>
```

C. Image a la une

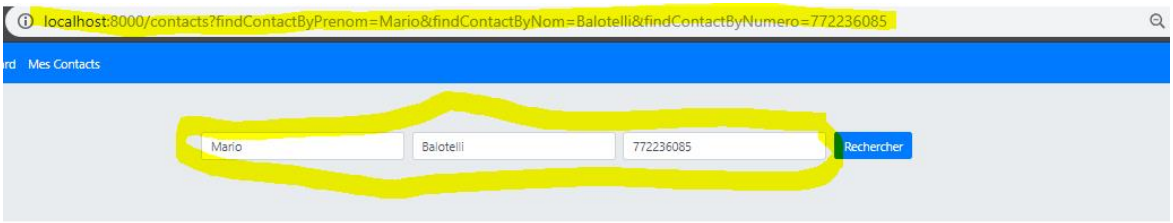
Dans mon tableau de bord on voit la liste des derniers contacts ajoutes avec des images.

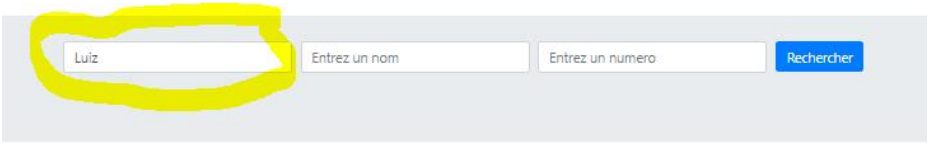


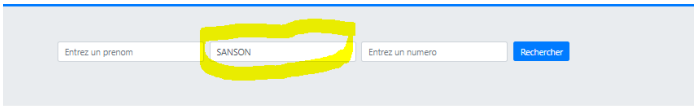

```
{% extends 'base' %}

{% block title 'My Contact' %}
{% block body %}

<div class="jumbotron text-center">
    <h1>My Contact</h1>
    <p>Bienvenue dans mon repertoire de contacts</p>
</div>
    <div class="container">
        <h2>Les derniers contacts ajoutés</h2>
        <div class="row flex">
            {% for contact in contacts  %}
                <div class="col-3">
                    <div class="card">
                        {% if contact.filename %}
                            <img src="{{ vich_uploader_asset(contact, 'imageFile') | imagine_filter('thumb')}}" alt="card-img-top" style="...">
                        {% else %}
                            <img src="/images/contacts/logo.png" alt="card-img-top" style="...">
                        {% endif %}
                        <div class="card-body">
                            <h5 class="card-title">
                                <a href=" {{ path('contact.show', {id: contact.id, slug: contact.slug}) }}">{{ contact.prenom }}</a>
                            </h5>
                            <p class="card-text" style="font-weight: bold;font-size: 1.5rem">+221 {{  contact.numero }}</p>
                        </div>
                    </div>
                </div>
            {% endfor %}
        </div>
    </div>
{% endblock %}
```

D. Filtrage des contacts

On pourra filtrer les contacts par leur nom, prénom, et/ou numéro



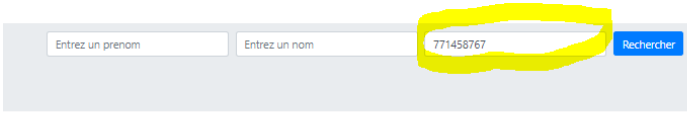




Mes Contacts

Morgan

+221 773876273



Mes Contacts

Boubacar

+221 771458767

```
<div class="jumbotron">
    <div class="container">
        {{ form_start(form) }}
        <div class="form-row">
            <div class="col">
                {{ form_row(form.findContactByPrenom) }}
            </div>
            <div class="col">
                {{ form_row(form.findContactByNom) }}
            </div>
            <div class="col">
                {{ form_row(form.findContactByNumero) }}
            </div>
            <div class="col">
                <button class="btn btn-primary">Rechercher</button>
            </div>
        </div>
        {{ form_end(form) }}
    </div>
</div>
```

```php
class ContactSearchType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add( child: 'findContactByPrenom', type: TextType::class,[
                'required'=>false,
                'label'=>false,
                'attr'=>[
                    'placeholder'=>'Entrez un prenom'
                ]
            ])
            ->add( child: 'findContactByNom', type: TextType::class,[
                'required'=>false,
                'label'=>false,
                'attr'=>[
                    'placeholder'=>'Entrez un nom'
                ]
            ])
            ->add( child: 'findContactByNumero', type: IntegerType::class,[
                'required'=>false,
                'label'=>false,
                'attr'=>[
                    'placeholder'=>'Entrez un numero'
                ]
            ])
        ;
    }
```

```php
     * @param ContactSearch $search
     * @return Query
     */
    public function findAllVisibleQuery(ContactSearch $search): Query
    {
        $query = $this->findVisibleQuery();

        if ($search->getFindContactByPrenom()){
            $query =$query
                ->andWhere('c.prenom = :findcontactprenom')
                ->setParameter( key: 'findcontactprenom',$search->getFindContactByPrenom());

        }
        if ($search->getFindContactByNom()){
        $query =$query
            ->andWhere('c.nom = :findcontactnom')
            ->setParameter( key: 'findcontactnom',$search->getFindContactByNom());

    }
        if ($search->getFindContactByNumero()){
        $query =$query
            ->andWhere('c.numero = :findcontactnumero')
            ->setParameter( key: 'findcontactnumero',$search->getFindContactByNumero());

    }
            return $query->getQuery();
    }
```