



Présentation du projet O'Galaxy

Titre Professionnel Développeur Web et Web Mobile

Soutenance Pierre Morin





Introduction

Qui suis-je ?

Parcours

2008 - 2016

Études supérieures

2016 - 2018

Community Manager / Rédacteur Web

2018 - 2021

Social Media Manager

2021 - 2023

Social Media Manager / Resp. Pôle Social Media

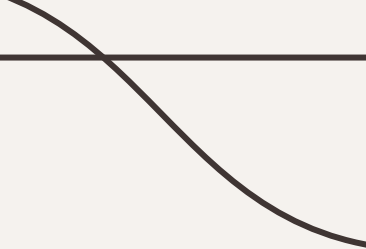


Pourquoi le développement web ?

- Curiosité présente depuis des années
- Volonté de découvrir la machinerie derrière les sites internet
- Impression d'avoir fait le tour de mon précédent métier
- Détermination, persévérance : des atouts pour la résolution de problèmes



Sommaire

- 
-
- 01. Équipe & présentation du projet**
 - 02. Conception**
 - 03. Développement : réalisations personnelles**
 - 04. Présentation des éléments de sécurité**
 - 05. Démonstration du projet**
 - 06. Veille**
 - 07. Conclusion**
-



Équipe & genèse du projet

Notre équipe



Pierre Morin

Scrum Master
Lead Dev Front



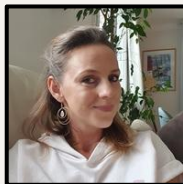
Jessica Troilo

Scrum Master
Lead Dev Front



Guillaume Ribas

Git Master
Product Owner



Agnès Julien

Lead Dev Back



Élodie Gauthier

Lead Dev Back

O'Galaxy, la genèse d'un projet

Pourquoi un site fictif ?

- 6 mois intense d'octobre à avril
- Envie de s'amuser sur le projet d'Apothéose
- Un projet "fun" dans des conditions professionnelles

Cibles du projet ?

- Membres de familles fortunées
- Célébrités



Organisation du temps de travail

- Organisation autour de la méthode **scrum**
- **4 sprints** d'une durée de 5 à 6 jours
- Gestion de projet via **Trello**
- Communication et réunions via **Discord**
- Lancement des journées par un **daily meeting**...
... et une petite partie de golf pour la cohésion !



Organisation des sprints

Sprint 2

Développement

Sprint 4

Relecture globale
Fix des bugs restants
Déploiement

01 — 02 — 03 — 04

Sprint 1

Réflexion sur le projet
Documents de conception

Sprint 3

Développement



Conception

User stories

- User story : phrase simple permettant de décrire une fonctionnalité
- 3 types d'utilisateurs :
 - les visiteurs
 - les utilisateurs connectés
 - les administrateurs

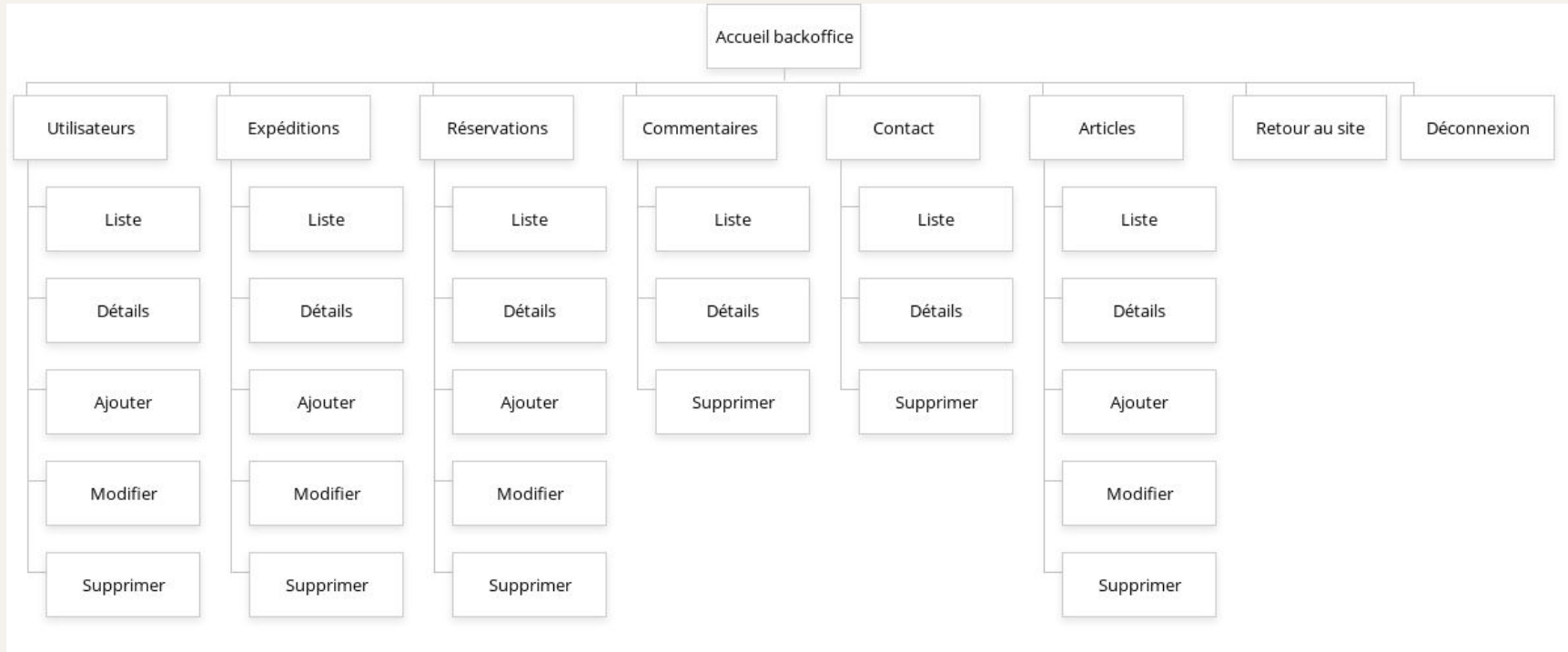
THEME	EN TANT QUE	JE SOUHAITE	AFIN DE...
Accueil	Visitor	Consulter la page d'accueil	Connaître les destinations phares
Expédition	Vistor	Consulter la liste de toutes les expéditions	Connaître l'ensemble des destinations
Expédition	Visitor	Consulter une expédition en particulier	Connaître les détails d'une expédition
Authentification	Visitor	Un formulaire d'inscription	Créer un compte client
Authentification	User	Un formulaire de connexion	Me connecter à mon compte
Expédition	User	Réserver une ou plusieurs expéditions	Partir en voyage
Réservation	User	Avoir accès à un formulaire de réservation	Réserver une expédition
Administration	Admin	Avoir accès à mon compte et au backoffice	Gérer mon site
Administration	Admin	Créer / Modifier / Supprimer des expéditions et activités	Tenir à jour le site
Administration	Admin	Gérer les Users	Autoriser / restreindre l'accès à certaines parties du site

Extrait des user stories

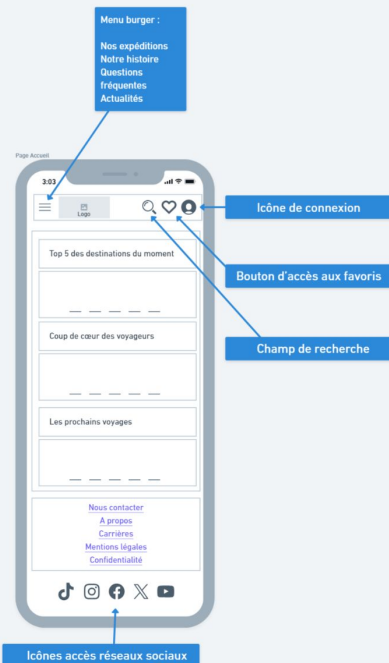
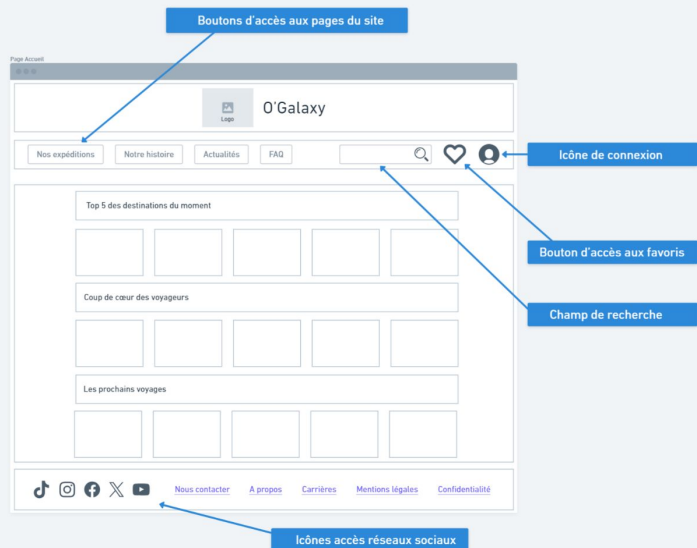
Arborescence du site



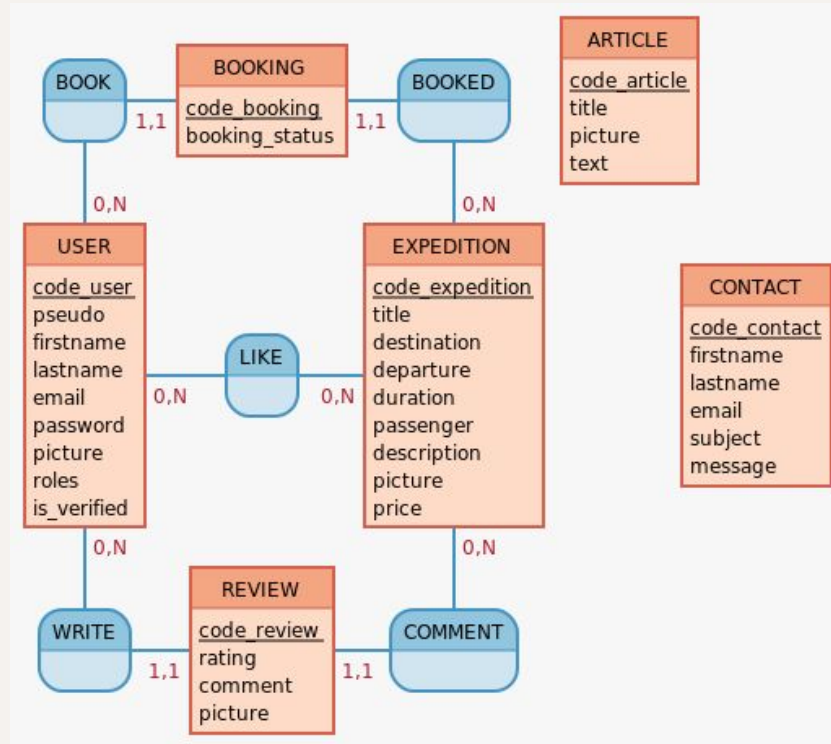
Arborescence du back office



Wireframes



MCD, Modèle Conceptuel de Données



MLD, Modèle Logique de Données

ARTICLE (code_article, title, picture, text)

BOOKING (code_booking, booking_status, #code_user, **#code_expedition**)

CONTACT (code_contact, firstname, lastname, email, subject, message)

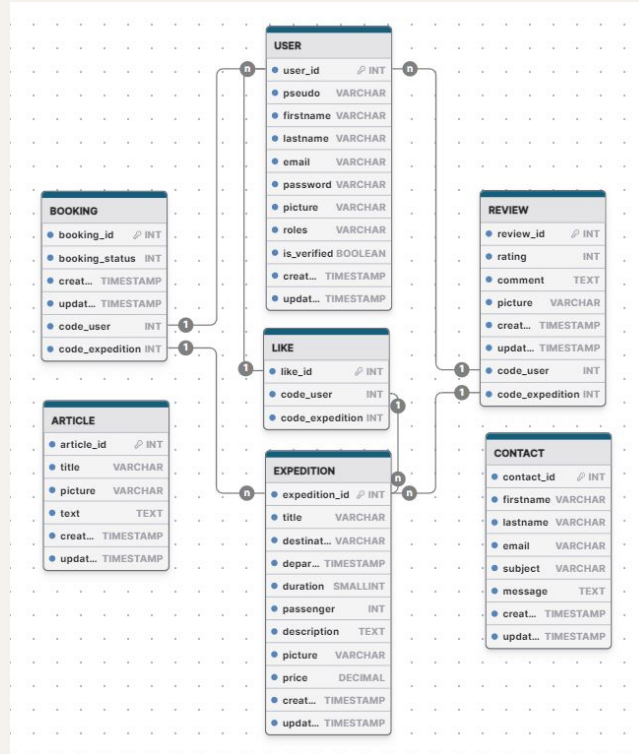
EXPEDITION (code_expedition, title, destination, departure, duration, passenger, description, picture, price)

LIKE (code_like, **#code_user**, **#code_expedition**)

REVIEW (code_review, rating, comment, picture, #code_user, **#code_expedition**)

USER (code_user, pseudo, firstname, lastname, email, password, picture, roles, is_verified)

MPD, Modèle Physique de Données



Dictionnaire de données

Table : `USER`

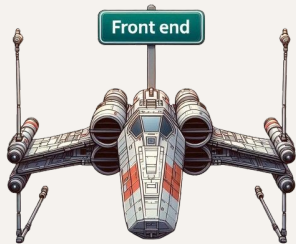
Colonne	Type	Description	Clé primaire	Auto-incrément	Non null
user_id	INT	Identifiant unique de l'utilisateur	Oui	Oui	Oui
pseudo	VARCHAR(255)	Pseudonyme de l'utilisateur	Non	Non	Oui
firstname	VARCHAR(255)	Prénom de l'utilisateur	Non	Non	Oui
lastname	VARCHAR(255)	Nom de famille de l'utilisateur	Non	Non	Oui
email	VARCHAR(255)	Adresse email de l'utilisateur	Non	Non	Oui
password	VARCHAR(255)	Mot de passe de l'utilisateur	Non	Non	Oui
picture	VARCHAR(255)	URL de la photo de profil de l'utilisateur	Non	Non	Non
roles	VARCHAR(255)	Rôles de l'utilisateur (ex. : admin, utilisateur)	Non	Non	Oui
is_verified	BOOLEAN	Indique si l'utilisateur est vérifié	Non	Non	Oui
created_at	TIMESTAMP	Date de création de l'utilisateur	Non	Non	Non
updated_at	TIMESTAMP	Date de mise à jour de l'utilisateur	Non	Non	Non

Table : `BOOKING`

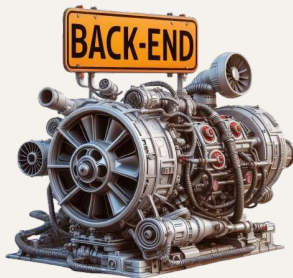
Colonne	Type	Description	Clé primaire	Auto-incrément	Non null
booking_id	INT	Identifiant unique de la réservation	Oui	Oui	Oui
booking_status	INT	Statut de la réservation (ex. : en attente, confirmé, annulé)	Non	Non	Oui
created_at	TIMESTAMP	Date de création de la réservation	Non	Non	Non
updated_at	TIMESTAMP	Date de mise à jour de la réservation	Non	Non	Non
code_user	INT	Identifiant de l'utilisateur ayant fait la réservation	Non	Non	Non
code_expedition	INT	Identifiant de l'expédition réservée	Non	Non	Non

Extraits du dictionnaire de données

Stack technique



CSS
Bootstrap
JavaScript
Twig



Symfony 7
MariaDB
MailJet



GitHub



Développement

Réalisations personnelles côté front-end

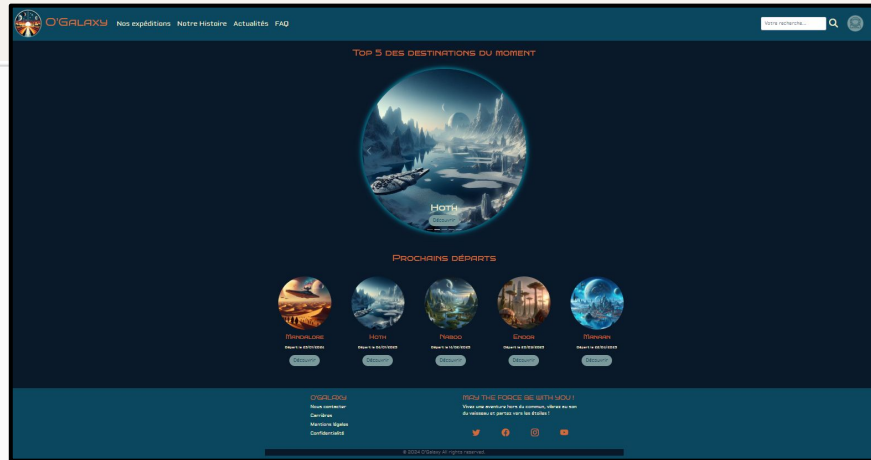
Durant ce projet de fin de formation, tous les membres de l'équipe venaient de la spécialisation Symfony !

When backend Developer does Frontend



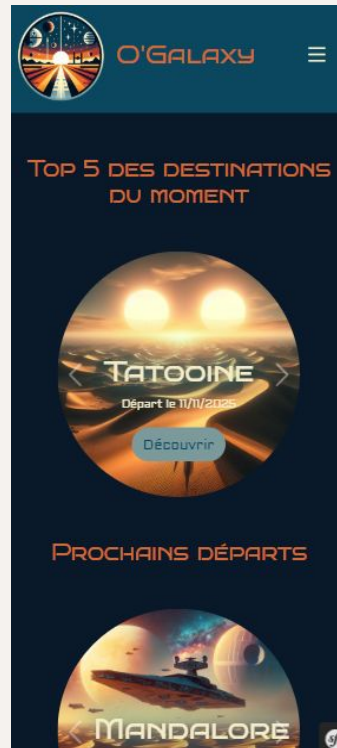
Réalisation front-end (statique)

```
<!-- Display the next departure - DESKTOP VERSION -->
<section id="next-departure-version-desktop" class="d-none d-lg-block">
  <h2 class="container pt-4 home__bigtitle mt-4 text-center mb-5">
    Prochains départs
  </h2>
  <div
    class="container d-flex gap-4 mt-3">
    <!-- CARD -->
    {% for currentHome in expeditionDeparture %}
      <div id="home__next__border" class="card d-flex mb-3 ps-5" style="width: 30rem;">
        
        <div class="card-body d-flex align-items-center flex-column home__card_bg">
          <h3 class="card-title home__title mb-3">
            {{ currentHome.title }}
          </h3>
          <p class="card-text home__text_dep">
            Départ le {{ currentHome.departure | date('d/m/Y') }}
          </p>
          <a href="{{ path('app_front_expedition_show', {id:currentHome.id}) }}" class="btn list_btn" role="button"
            aria-label="Bouton pour afficher l'expédition choisie">
            Découvrir
          </a>
        </div>
      </div>
    {% endfor %}
  </div>
</section>
```

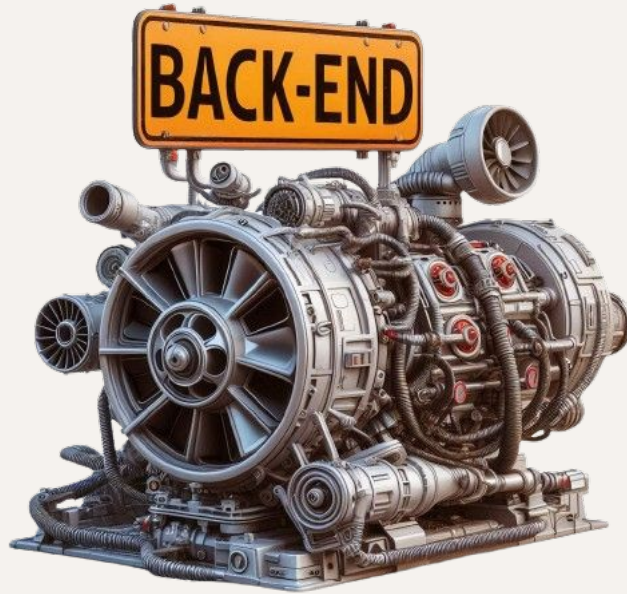


Réalisation front-end (partie dynamique)

```
<!-- Display the next departure - MOBILE VERSION -->
<section id="next-departure-version-mobile" class="d-block d-lg-none">
  <h2 class="container pt-4 home__bigtitle mt-4 text-center mb-5">
    Prochains départs
  </h2>
  <div id="carouselNextDeparture" class="container d-flex justify-content-center carousel carousel-dark slide">
    <div class="carousel-inner w-75">
      <% for currentHome in expeditionDeparture %>
        <div class="carousel-item {% if loop.first %}active{% endif %}" data-bs-interval="10000">
          
          <div class="carousel-caption">
            <h5 id="home__title">
              {{ currentHome.title }}
            </h5>
            <p class="home__text_dep">
              Départ le {{ currentHome.departure | date ('d/m/Y') }}
            </p>
            <a href="{{ path('app_front_expedition_show', {id:currentHome.id}) }}" class="btn list_btn" role="button"
              aria-label="Bouton pour afficher le slide choisi">
              Découvrir
            </a>
          </div>
        </div>
      </for>
    </div>
    <div class="carousel-control-prev white__arrow" type="button" data-bs-target="#carouselNextDeparture"
      data-bs-slide="prev" aria-label="Bouton pour afficher sur le slide précédent">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="visually-hidden">
        Previous
      </span>
    </div>
    <div class="carousel-control-next white__arrow" type="button" data-bs-target="#carouselNextDeparture"
      data-bs-slide="next" aria-label="Bouton pour afficher sur le slide suivant">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="visually-hidden">
        Next
      </span>
    </div>
  </div>
</section>
```



Réalisation personnelle côté back-end



Réalisations personnelles côté back-end

Composant métier : mise en place de
l'authentification et de la gestion
des utilisateurs



Réalisations personnelles côté back-end

- Installation des packages importants :
security-bundle
maker-bundle
- Configuration du fichier **security.yaml**
dans **config/packages**

```
security:
    # https://symfony.com/doc/current/security.html#registering-the-user-hashing-passwords
    password_hashers:
        Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'
    # https://symfony.com/doc/current/security.html#loading-the-user-the-user-provider
    providers:
        # used to reload user from session & other features (e.g. switch_user)
        app_user_provider:
            entity:
                class: App\Entity\User
                property: email
    firewalls:
        dev:
            pattern: ^/(_(profiler|wdt)|css|images|js)/
            security: false
    main:
        lazy: true
        provider: app_user_provider
        custom_authenticator: App\Security\LoginFormAuthenticator
        logout:
            path: app_logout
        remember_me:
            secret: '%kernel.secret%'
            lifetime: 604800
            path: /

    # Easy way to control access for large sections of your site
    # Note: Only the *first* access control that matches will be used
    access_control:
        - { path: ^/expedition/favorites, roles: ROLE_USER }
        - { path: ^/admin, roles: [ROLE_ADMIN] }

when@test:
    security:
        password_hashers:
            # By default, password hashers are resource intensive and take time. This is
            # important to generate secure password hashes. In tests however, secure hashes
            # are not important, waste resources and increase test times. The following
            # reduces the work factor to the lowest possible values.
            Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:
                algorithm: auto
                cost: 4 # Lowest possible value for bcrypt
                time_cost: 3 # Lowest possible value for argon
                memory_cost: 10 # Lowest possible value for argon
```

Réalisations personnelles côté back-end

- Création de l'entité **User** à l'aide du maker-bundle

```
$ php bin/console make:user
The name of the security user class (e.g. User) [User]:
> User

Do you want to store user data in the database (via Doctrine)? (yes/no) [yes]:
> yes

Enter a property name that will be the unique "display" name for the user (e.g. email,
username, uuid) [email]:
> email

Will this app need to hash/check user passwords? Choose No if passwords are not needed or will
be checked/hashed by some other system (e.g. a single sign-on server).

Does this app need to hash/check user passwords? (yes/no) [yes]:
> yes

created: src/Entity/User.php
created: src/Repository/UserRepository.php
updated: src/Entity/User.php
updated: config/packages/security.yaml
```

Réalisations personnelles côté back-end

- L'entité **User** est créée et implémente **UserInterface** et **PasswordAuthenticatorUserInterface**

```
<?php

namespace App\Entity;

use DateTimeImmutable;
use Doctrine\ORM\Mapping as ORM;
use App\Repository\UserRepository;
use Doctrine\Common\Collections\Collection;
use Doctrine\Common\Collections\ArrayCollection;
use Symfony\Component\Security\Core\User\UserInterface;
use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;
use Symfony\Component\Security\Core\User>PasswordAuthenticatedUserInterface;
use Symfony\Component\Validator\Constraints as Assert;

#[ORM\Entity(repositoryClass: UserRepository::class)]
#[ORM\UniqueConstraint(name: 'UNIQ_IDENTIFIER_EMAIL', fields: ['email'])]
#[UniqueEntity(fields: ['email'], message: 'Il existe déjà un compte relié à cette adresse')]
#[ORM\HasLifecycleCallbacks]
class User implements UserInterface, PasswordAuthenticatedUserInterface
{
    ...
}
```

Réalisations personnelles côté back-end

- Mise à jour de la base de données :

php bin/console make:migration

php bin/console doctrine:migrations:migrate

Réalisations personnelles côté back-end

- Création d'un formulaire d'authentification :
php bin/console make:auth

```
public function onAuthenticationSuccess(Request $request, TokenInterface $token, string $firewallName): ?Response
{
    if ($targetPath = $this->getTargetPath($request->getSession(), $firewallName)) {
        return new RedirectResponse($targetPath);
    }

    // For example:
    return new RedirectResponse($this->urlGenerator->generate('app_home'));
    // throw new \Exception('provide a valid redirect inside '.__FILE__');
}
```

```
$ php bin/console make:auth
What style of authentication do you want? [Empty authenticator]:
[0] Empty authenticator
[1] Login form authenticator
> 1

The class name of the authenticator to create (e.g. AppCustomAuthenticator):
> LoginFormAuthenticator

Choose a name for the controller class (e.g. SecurityController) [SecurityController]:
> SecurityController

Do you want to generate a '/logout' URL? (yes/no) [yes]:
> yes

created: src/Security/LoginFormAuthenticator.php
updated: config/packages/security.yaml
created: src/Controller/SecurityController.php
created: templates/security/login.html.twig

Success!

Next:
- Customize your new authenticator.
- Finish the redirect "TODO" in the
App\Security\LoginFormAuthenticator::onAuthenticationSuccess() method.
- Review & adapt the login template: templates/security/login.html.twig.
```


Réalisations personnelles côté back-end

- Affichage du formulaire de connexion : customisation du formulaire dans le template Twig pour l'affichage de la vue

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

{% block title %}Connexion{% endblock %}

{% block body %}
<!-- Display the login -->
<section id="login" aria-labelledby="to-log-in">
  <div class="container col-12 col-lg-6 my-4">
    <h3 id="list_bigtitle" class="display-4">Se connecter</h3>
    <form method="post" novalidate>
      {% if error %}
      <div class="alert alert-danger" role="alert">{{ error.messageKey|trans(error.messageData, 'security') }}</div>
      {% endif %}

      {% if app.user %}
      <div class="login_color mb-3">
        Bienvenue {{ app.user.userIdentifier }}, <a href="{{ path('app_logout') }}" role="button"
        aria-label="Bouton pour se déconnecter">Déconnexion</a>
      </div>
      {% endif %}

      <h2 class="h3 mb-3 font-weight-normal login_color">Veuillez vous connecter</h2>
      <div class="col-8">
        <label for="inputEmail" class="login_color">Adresse mail</label>
        <input type="email" value="{{ last_username }}" name="email" id="inputEmail" class="form-control mb-3"
        autocomplete="email" required autofocus>
        <label for="inputPassword" class="login_color">Mot de passe</label>
        <input type="password" name="password" id="inputPassword" class="form-control mb-3"
        autocomplete="current-password" required>

        <input type="hidden" name="_csrf_token" value="{{ csrf_token('authenticate') }}">
      </div>
      <div class="checkboxbox mt-3 mb-3">
        <label class="login_color">
          <input type="checkbox" name="_remember_me" role="button"
          aria-label="Bouton pour mémoriser sa connexion"> Se souvenir de moi</label>
        </div>
      <button class="btn login_btn" type="submit" aria-label="Bouton pour se connecter">Connexion</button>
    </form>
    <h2 class="h3 mt-5 mb-3 font-weight-normal login_color">Je ne possède pas de compte</h2>
    <button class="btn login_btn" onclick="window.location.href='{{ path('app_register') }}'"
    aria-label="Bouton pour se créer un compte">Créer un compte</button>
  </div>
</section>
{% endblock %}
```

Réalisations personnelles côté back-end

Composants d'accès aux données :

- Symfony : framework fonctionnant sur le design pattern MVC
- MVC : Modèle - Vue - Contrôleur



Réalisations personnelles côté back-end

- Création d'une route dans le **UserController**
- Récupération des données via le **UserRepository**
- Transmission des données à la vue

```
<?php

namespace App\Controller\Back;

use App\Entity\User;
use App\Form\UserType;
use App\Repository\UserRepository;
use Doctrine\ORM\EntityManagerInterface;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Attribute\Route;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

#[Route('/admin/utilisateur', name: 'app_admin_user_')]
class UserController extends AbstractController
{
    /**
     * Method to show the list of all users
     *
     * @param UserRepository $userRepository
     * @return Response
     */

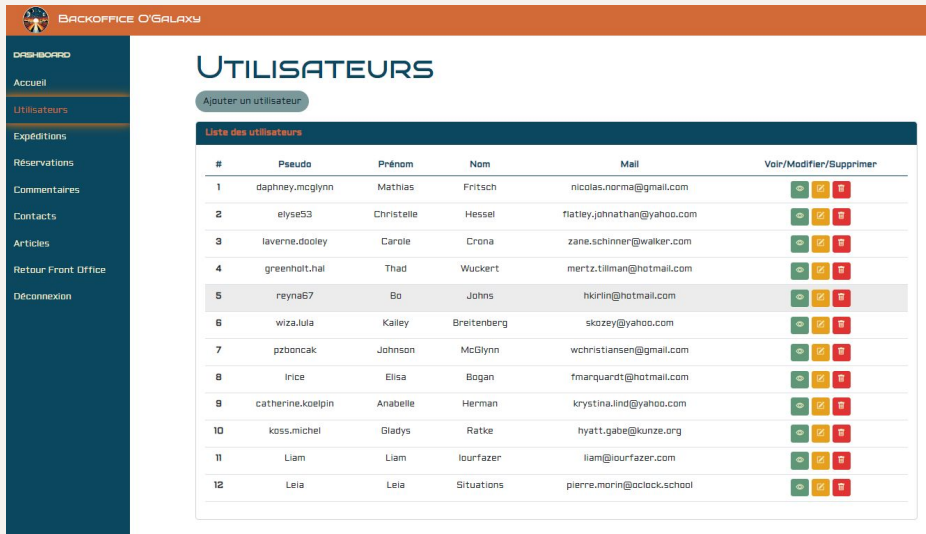
    #[Route('/', name: 'list', methods: ['GET'])]
    public function list(UserRepository $userRepository): Response
    {
        $allUsers = $userRepository->findAll();

        return $this->render('back/user/list.html.twig', [
            'userList' => $allUsers
        ]);
    }
}
```

Réalisations personnelles côté back-end

- Création d'un template pour afficher la vue

```
{% for currentUser in userList %}
<tr class="text-center">
<th scope="row">
    {{ currentUser.id }}
</th>
<td>
    {{ currentUser.pseudo }}
</td>
<td>
    {{ currentUser.firstname }}
</td>
<td>
    {{ currentUser.lastname }}
</td>
<td>
    {{ currentUser.email }}
</td>
...
</tr>
{% endfor %}
```



#	Pseudo	Prénom	Nom	Mail	Voir/Modifier/Supprimer
1	daphney.mcglynn	Mathias	Fritsch	nicolas.normai@gmail.com	Voir Modifier Supprimer
2	elyse53	Christelle	Hessel	fiatley.johnathan@yahoo.com	Voir Modifier Supprimer
3	laverne.dooley	Carole	Crona	zane.schinner@walker.com	Voir Modifier Supprimer
4	greenholt.hai	Thad	Wuckert	mertz.tillman@hotmail.com	Voir Modifier Supprimer
5	reyna67	Bo	Johns	hkirlin@hotmail.com	Voir Modifier Supprimer
6	wiza.jula	Kailey	Breitenberg	skozey@yahoo.com	Voir Modifier Supprimer
7	pzboncak	Johnson	McGlynn	wchristiansen@gmail.com	Voir Modifier Supprimer
8	lrice	Elisa	Bogan	fmarquardt@hotmail.com	Voir Modifier Supprimer
9	catherine.koelpin	Anabelle	Herman	krystina.lind@yahoo.com	Voir Modifier Supprimer
10	koss.michel	Gladys	Ratke	hyatt.gabe@kunze.org	Voir Modifier Supprimer
11	Liam	Liam	lourfazer	liam@lourfazer.com	Voir Modifier Supprimer
12	Leia	Leia	Situations	pierre.morin@oclock.school	Voir Modifier Supprimer



Présentation des éléments de sécurité

La sécurité, un élément essentiel

- La sécurité, composante essentielle pour tout site web ou application
- Protéger les données sensibles
- Conformité légale : RGPD en Europe
- Éviter les attaques courantes : injections SQL, XSS, CSRF...

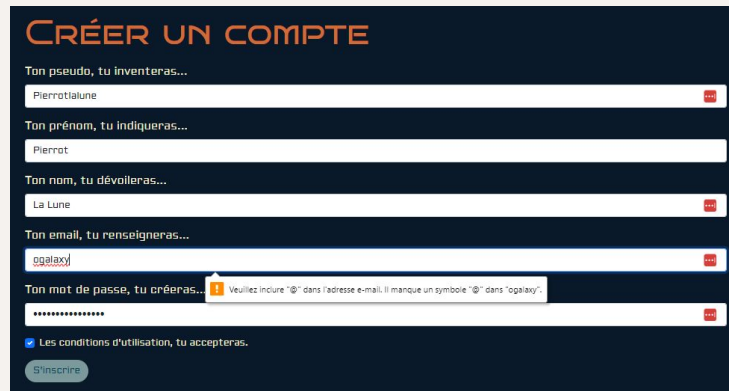


La sécurité sur Symfony

- Security Component : configuration du fichier security.yaml
- Protection CSRF par défaut sur les formulaires avec Form Component
- Protection injections SQL : ORM Doctrine utilise des requêtes préparées
- Protection XSS : échappement automatique avec Twig

La sécurité sur Symfony

- Hashage des mots de passe : mot de passe hashé en BDD et comparaison avec le mot de passe en clair lors de la connexion
- Validator Component : valider et filtrer les données, utilisation de contraintes prédéfinies



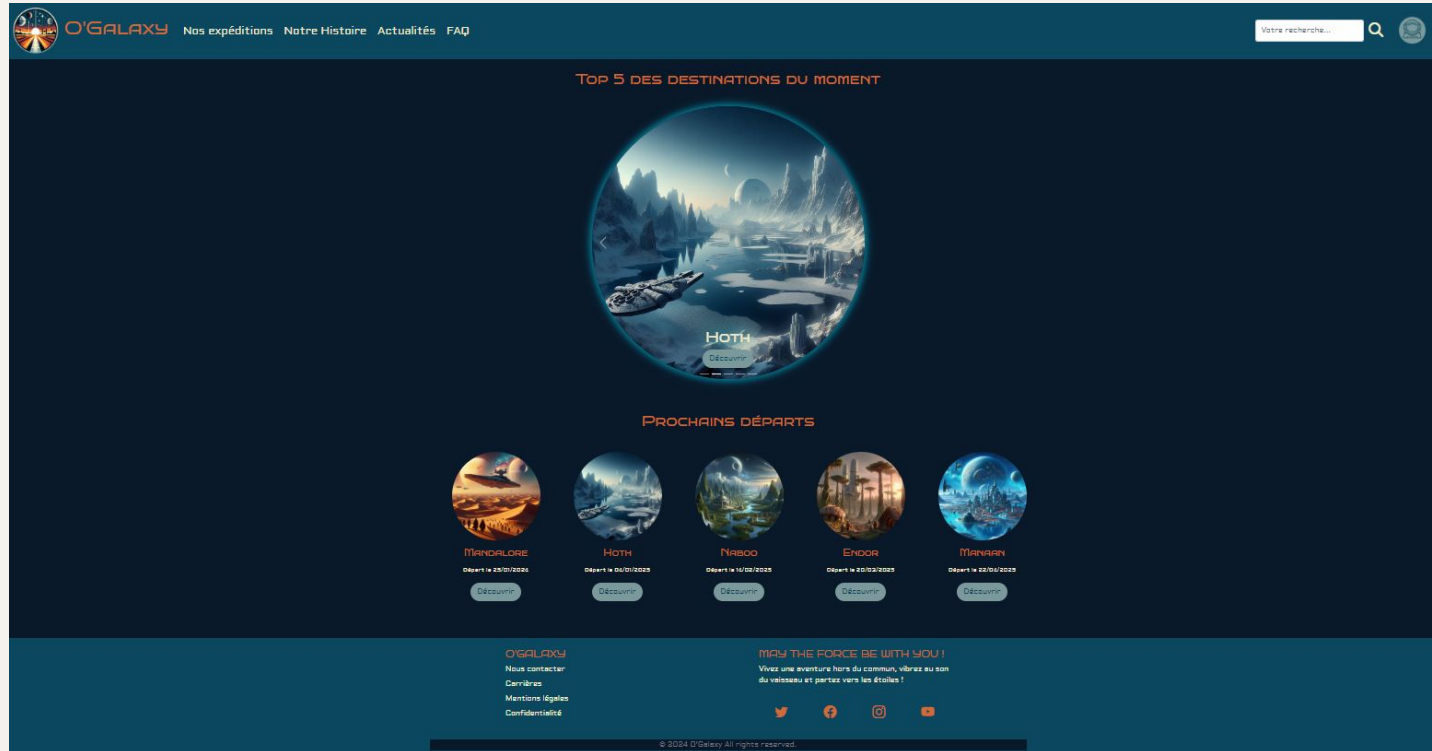
The screenshot shows a registration form titled "CRÉER UN COMPTE" on a dark blue background. The form contains several input fields with labels and validation messages:

- Ton pseudo, tu inventeras...**: Input field with "Pierrotialune" and a red error icon.
- Ton prénom, tu indiqueras...**: Input field with "Pierrot".
- Ton nom, tu dévoileras...**: Input field with "La Lune" and a red error icon.
- Ton email, tu renseigneras...**: Input field with "ogalaxy" and a red error icon.
- Ton mot de passe, tu créeras...**: Input field with masked characters "*****". A tooltip message says: "Veillez inclure 'Q' dans l'adresse e-mail. Il manque un symbole 'Q' dans 'ogalaxy'".
- Les conditions d'utilisation, tu accepteras.**: A checked checkbox.
- S'inscrire**: A green button.



Démonstration du projet

Démo création d'un compte O'Galaxy





Veille

Veille technologique & sectorielle

- Documentation des langages et des frameworks
- Veille Twitter dédiée au développement web
- Diverses chaînes YouTube : Grafikart, Melvynx, Nouvelle Techno...
- Blogs : CodeGarage.fr, SymfonyCasts, dev.to...



Veille sécurité

- Documentation Security de Symfony
- Blog de l'ANSSI & alertes sécurité CERT-FR
- Diverses chaînes YouTube : Micode, Underscore_, Pour1nf0, Hafnium...
- Site hacksplaining



Conclusion

Conclusion

- Une super ambiance d'équipe !
 - Fierté d'avoir pu participer à ce projet
 - Confirmation de mon appétence pour le développement web
 - Envie d'aller découvrir un framework front-end avec un projet personnel en React
-

