

DOCUMENTATION

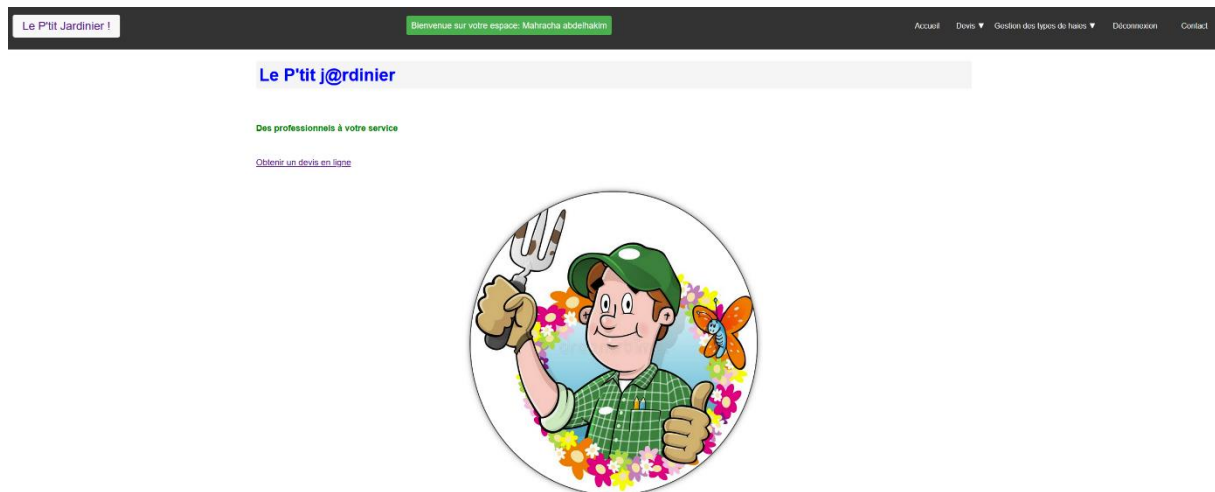
Le P'tit j@rdinier

1. Introduction

Ce document fournit une description détaillée du projet de développement d'une application web avec le Framework **SYMFONY** permettant de calculer le tarif estimé de l'entretien des jardins et de gérer les informations des clients.

2. Détails du Projet

Nom du Projet : **Le P'tit j@rdinier**



Description

Cette application est créée pour faciliter aux particuliers et aux organisations de l'entretien des jardins d'obtenir un tarif estimé. Elle permet également de consulter et modifier les informations des clients ainsi que de voir les types de haies.

Objectifs

- Créer une application web pour calculer le tarif estimé de l'entretien des jardins.
- Permettre aux utilisateurs de consulter et modifier les informations des clients.
- Afficher les types de haies disponibles pour l'entretien.

Résultats

L'application est fonctionnelle et permet aux utilisateurs de calculer le tarif estimé de l'entretien des jardins, de consulter et modifier les informations des clients, et de voir les types de haies.

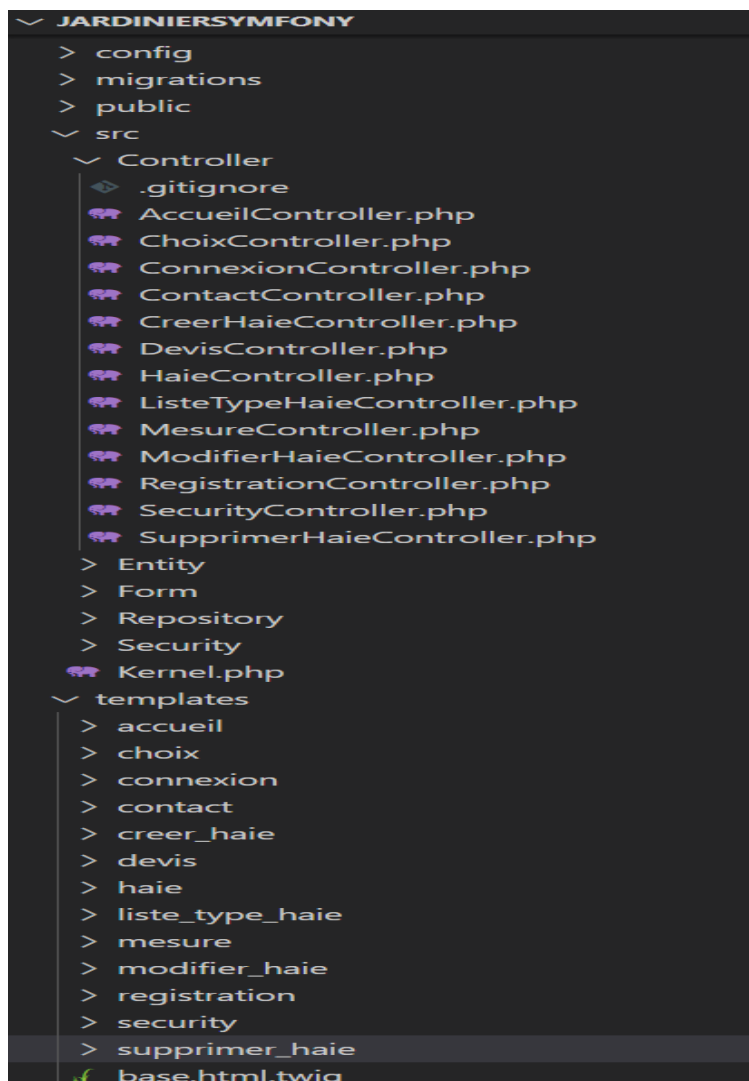
Durée

3 mois

3. Technologies Utilisées

- **HTML5** : Structure de base des pages web.
- **CSS3** : Styles et mise en page des pages web.
- **PHP** : Logique de serveur pour le traitement des données.
- **MySQL** : Base de données pour stocker les informations des clients et des tarifs.
- **Bootstrap** : Framework CSS pour des interfaces réactives et élégantes.
- **Symfony** : Framework PHP pour la gestion des fonctionnalités de l'application.

4. Structure du Projet



BDD : MySQL

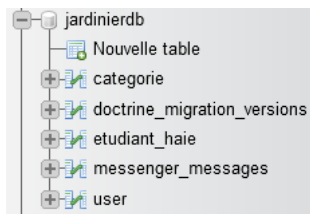


Table categorie :

					id	libelle
<input type="checkbox"/>		Éditer		Copier		Supprimer
					1	Persistent
<input type="checkbox"/>		Éditer		Copier		Supprimer
					2	Caducue

Table etudiant_haie :


<div><div>←T→</div><div></div></div>						id	code	nom	prix	categorie_id
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	1	LA	Laurier	40.00	1		
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	6	SA	SARA	58.00	1		
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	8	HA	HAKIM	12.00	2		
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	22	NA	Naira	15.00	2		

Table user :

<div><div><div><div></div><div></div><div></div></div><div></div><div></div></div></div>												id	email	prenom	nom	roles	password
<div><div><div></div></div><div><div></div><div></div><div></div></div><div><div>Éditer</div><div>Copier</div><div>Supprimer</div></div></div>	1	hakim@gmail.com	abdelhakim	Mahracha	<div></div>	\$2y\$13\$OdT04osZ9ptqm0MWfJcL.VoEde90Yyl5BjJOU16zr...											
<div><div><div></div></div><div><div></div><div></div><div></div></div><div><div>Éditer</div><div>Copier</div><div>Supprimer</div></div></div>	2	lara@gmail.com	lara	flores	<div></div>	\$2y\$13\$m2PbvO3JPfTydgJMLNx.9OT7Focw8Mgjp53im9PWFu4...											
<div><div><div></div></div><div><div></div><div></div><div></div></div><div><div>Éditer</div><div>Copier</div><div>Supprimer</div></div></div>	3	clarafer@gmail.com	Clara	Fernandez	<div></div>	\$2y\$13\$yhwg0T5zIOYJF2W0GkXKucKRSRk6.ItBCKwUCptBqv...											

Pour exécuter le projet en symfony :

- On démarre ce serveur web avec la commande :

```
symfony server:start
//ou bien
symfony serve
```

Création de la base de données

- `php bin/console doctrine:database:create`

Création d'une entité

- `php bin/console make:entity`

L'utilisateur : Définir une entité User

Utilisez la commande dédiée `make:user` pour créer l'entité au lieu de la commande traditionnelle `make:entity`

- `$ symfony console make:user`

La connexion : Configurer le système d'authentification

Utilisons la commande `$ symfony console make:auth` pour :

- mettre à jour la configuration de sécurité,
- générer un template pour la connexion
- et créer une classe d'authentification (authenticator)



L'inscription : Créer un formulaire d'inscription

- `symfony console make:registration-form`



Faire la migration

- `php bin/console make:migration`
- `php bin/console doctrine:migrations:migrate`

Enregistrer les données dans BDD

- `use App\Entity\EtudiantHaie;`

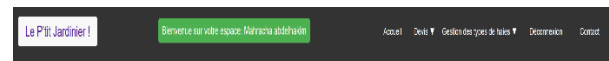
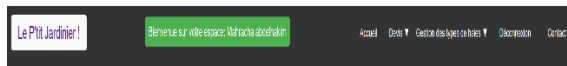
Création du contrôleur et de la vue

- `symfony console list make`
- `symfony console make:controller`

5. Fonctionnalités

Calcul du Tarif d'Entretien

- Formulaire permettant aux utilisateurs de saisir les détails de leur jardin pour obtenir un tarif estimé.



- Créer le contrôleur et la vue de choix du type utilisateur :

```
class ChoixController extends AbstractController
{
    #[Route('/choix', name: 'app_choix')]
    public function index(): Response
    {
        // Récupération de la valeur de 'userType' depuis la requête
        $request = Request::createFromGlobals();
        $userType = $request->get('userType');

        return $this->render('choix/index.html.twig', [
            'controller_name' => 'ChoixController',
        ]);
    }
}
```

Vue :

- `action="{{ path('app_mesure') }}"` permet de nous diriger vers la page avec la route /mesure

```
<form action="{{ path('app_mesure') }}" method="post" class="custom-form">
    <div class="form-group">
        <label for="userType" class="user-label"><b>Vous
êtes:</b></label>
```

```

        <div class="radio-buttons">
            <div class="form-group radio-container">
                <input type="radio" id="particulier"
name="userType" value="particulier" checked>
                <label for="particulier">Particulier</label>
            </div>
            <div class="form-group radio-container">
                <input type="radio" id="entreprise"
name="userType" value="entreprise">
                <label for="entreprise">Entreprise</label>
            </div>
        </div>
    </div>
    <button type="submit" class="btn btn-
green">Valider</button>
</form>

```

➤ Créer le contrôleur et la vue de mesure:

```

class MeasureController extends AbstractController
{
    #[Route('/mesure', name: 'app_mesure')]
    public function index(Request $request, SessionInterface
$session, EtudiantHaieRepository $haieRepository): Response
    {
        //la liste des haies
        $lesHaies = $haieRepository->findAll();

        // Récupérer le type d'utilisateur à partir des données du
formulaire
        $typeUtilisateur = $request->request->get('typeUtilisateur');

        // Créer une variable de session pour stocker le type
d'utilisateur
        $session->set('typeUtilisateur', $typeUtilisateur);

        return $this->render('mesure/index.html.twig', [
            'controller_name' => 'MeasureController',
            'lesHaies' => $lesHaies,
        ]);
    }
}

```

Vue :

- `action="{{ path('app_devis') }}"` permet de nous diriger vers la page avec la route `/devis`

```
<form method="post" action="{{ path('app_devis') }}" class="custom-form">
    <div class="form-group">
        <label for="typeHaie" class="form-label">Choisir
votre type de haie:</label>
        <select id="typeHaie" name="typeHaie" class="form-control">
            {% for haie in lesHaies %}
                <option value="{{ haie.getNom() }}">{{
haie.getNom() }}</option>
            {% endfor %}
        </select>
    </div>
    <div class="form-group">
        <label for="longueur" class="form-label">Longueur:</label>
        <input type="text" id="longueur" name="longueur"
class="form-control">
    </div>
    <div class="form-group">
        <label for="hauteur" class="form-label">Hauteur:</label>
        <input type="text" id="hauteur" name="hauteur"
class="form-control">
    </div>
    <button type="submit" class="btn btn-green">Suivant</button>
</form>
```

Gestion des Clients

- Affichage des informations des clients.

Types de Haies

- Affichage des différents types de haies disponibles.

LISTE TYPE DE HAIES:

Code	Nom	Prix	Actions
LA	Laurier	40.00	Consultation/Modification
SA	SARA	58.00	Consultation/Modification
HA	HAKIM	12.00	Consultation/Modification
NA	Naira	15.00	Consultation/Modification

Code du contrôleur :

- **\$haies = \$haieRepository->findAll();** c'est pour récupérer tous la liste des haies de la base de donnée.

```

class ListeTypeHaieController extends AbstractController
{
    #[Route('/liste/type/haie', name: 'app_liste_type_haie')]
    public function index(Request $request, EntityManagerInterface
$entityManager, EtudiantHaieRepository $haieRepository): Response
    {
        $haie = new EtudiantHaie();
        $form = $this->createForm(HaieType::class, $haie);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $entityManager->persist($haie);
            $entityManager->flush();
        }

        $haies = $haieRepository->findAll();
        return $this->render('liste_type_haie/index.html.twig', [
            'haies' => $haies,
            'form' => $form->createView(),
        ]);
    }
}

```

Vue :

- **href="{ path('modifier_haie', { 'code': haie.getCode() }) }"** permet de se diriger vers le chemin modifier_haie/+ **code choisi** à consulter ou modifier

```

<table>
    <tr>
        <th>Code</th>

```



```

        <th>Nom</th>
        <th>Prix</th>
        <th>Actions</th>
    </tr>
    {% for haie in haies %}
    <tr>
        <td>{{ haie.code }}</td>
        <td>{{ haie.nom }}</td>
        <td>{{ haie.prix }}</td>
        <td>
            <a href="{{ path('modifier_haie', { 'code': haie.getCode()
    }) }}">
                Consultation/Modification
            </a>
        </td>
    </tr>
    {% endfor %}
</table>

```

- Ajouter un type de haie.

Le P'tit Jardinier !

Bienvenue sur votre espace: Mahracha abdelhakim

[Accueil](#)
[Devis ▼](#)
[Gestion des types de haies ▼](#)
[Déconnexion](#)
[Contact](#)

AJOUTER UN TYPE DE HAIE

Code

Nom

Prix

Categorie haie

Persistant ▼

Enregistrer

Code du contrôleur :

```

class CreerHaieController extends AbstractController
{
    #[Route('/creerhaie', name: 'app_creer_haie')]
    public function index(Request $request, EntityManagerInterface $entityManager, EtudiantHaieRepository $haieRepository): Response
    {
    }
}

```

```

$haie = new EtudiantHaie();
$form = $this->createForm(HaieType::class, $haie);

$form->handleRequest($request);

if ($form->isSubmitted() && $form->isValid()) {

    $entityManager->persist($haie);
    $entityManager->flush();

    // Ajoute un message flash
    $this->addFlash('success', 'Haie ajouté avec succès!');

    return $this->redirectToRoute('app_creer_haie');

}

$lesHaies = $haieRepository->findAll();

return $this->render('creer_haie/index.html.twig', [
    'form' => $form->createView(),
]);
}
}

```

- Générer le formulaire avec la commande : `php bin/console make:form`

```

public function buildForm(FormBuilderInterface $builder, array
$options): void
{
    $builder
        ->add('code')
        ->add('nom')
        ->add('prix')
        ->add('categorie', EntityType::class, [
            'label' => 'Categorie haie',
            'class' => Categorie::class,
            'choice_label' => 'libelle',
        ])
}

```

```

        ->add('ENREGISTRER', SubmitType::class)

    ;

}

```

Vue :

```

{{ form_start(form, {'attr': {'class': 'styled-form'}}) }}
    <div class="form-group">
        {{ form_label(form.code) }}
        {{ form_widget(form.code, {'attr': {'class': 'form-control'}}) }}
    </div>
    <div class="form-group">
        {{ form_label(form.nom) }}
        {{ form_widget(form.nom, {'attr': {'class': 'form-control'}}) }}
    </div>
    <div class="form-group">
        {{ form_label(form.prix) }}
        {{ form_widget(form.prix, {'attr': {'class': 'form-control'}}) }}
    </div>
    <div class="form-group">
        {{ form_label(form.categorie) }}
        {{ form_widget(form.categorie, {'attr': {'class': 'form-control'}}) }}
    </div>
    <div class="form-group">
        {{ form_widget(form.ENREGISTRER, {'attr': {'class': 'btn btn-primary'}}) }}
    </div>
{{ form_end(form) }}

```

- Consultation, modification et suppression des différents types de haies disponibles pour l'entretien.

Le P'tit Jardinier !

Bienvenue sur votre espace: Mahracha abdelhakim

[Accueil](#)
[Devis ▼](#)
[Gestion des types de haies ▼](#)
[Déconnexion](#)
[Contact](#)

MODIFIER HAIE

Code

LA

Nom

Laurier

Prix

40

Categorie

Persistant ▼

Enregistrer

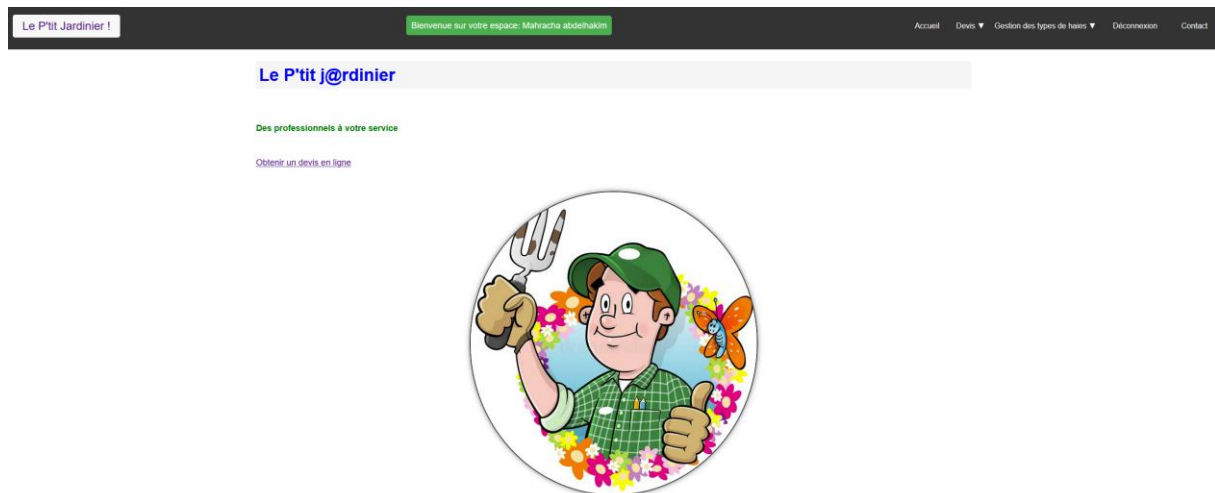
Supprimer

Retour

6. Interfaces Utilisateurs

Page d'Accueil

La page d'accueil affiche les informations générales de l'application et permet l'accès aux fonctionnalités de calcul de tarif, de gestion des clients, et de visualisation des types de haies.



8. Conclusion

Ce projet de développement d'une application web pour "Le P'tit j@rdinier" permet de simplifier le processus d'obtention de tarifs pour l'entretien des jardins et de gestion des informations des clients. Grâce à l'utilisation de technologies telles que **HTML5, CSS3, PHP, MySQL, Bootstrap et Symfony**, l'application offre une expérience utilisateur fluide et efficace.