

Projet 11

Répondre aux incidents et aux demandes d'assistance et d'évolution

Traiter des demandes concernant les applications

Projet Maya (en cours de formation, travail en équipe)

Contexte

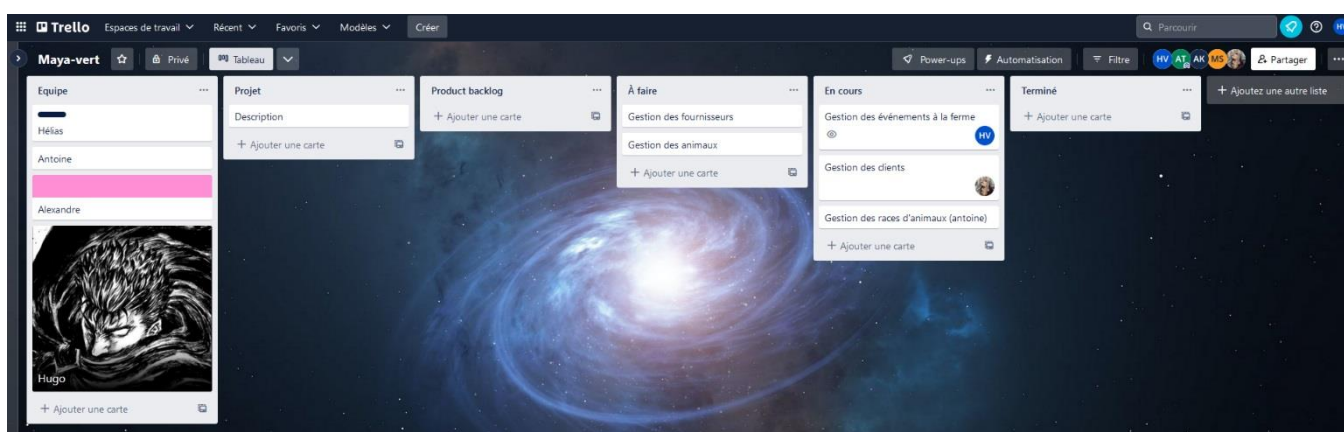
La Ferme Maya propose des fruits et légumes de saison issus d'une agriculture raisonnée ou biologique. Les clients de la ferme peuvent composer leur propre panier ou choisir parmi les paniers de la semaine. Leurs achats sont mis à disposition à la Ferme Maya le jour de leur choix.

Le site vitrine de la ferme est ancien et ne donne plus satisfaction. La ferme fait appel à notre société pour effectuer la refonte du site.

Prise en charge de la demande par le prestataire

Le travail est divisé en équipes, chacune travaillant sur une étape.

Nous avons créé un tableau « **Trello** » afin de répartir les tâches et leur suivi.



La mission globale de mon équipe :

Gérer les événements à la ferme, les clients, les fournisseurs, les races d'animaux et les animaux.

Je suis en charge de la partie « **évènements** »

Présentation actuelle du menu :



The image shows a green navigation bar for a website titled 'La ferme de Maya'. The menu items are: Accueil, Produits, Recettes, Clients, Commandes, Statistiques (with a dropdown arrow), and Se connecter.

Il s'agit d'ajouter une rubrique *évènements* au menu de l'application.

Service demandé

La rubrique événements doit se composer ainsi:

titre,
description,
date,
horaires

Outils utilisés

Symfony

Symfony est un des framework les plus prisés du web, qui, plus qu'un simple support de travail, est une technologie alliant rapidité de développement, sécurité, modularité et évolutivité du code. C'est un framework PHP open Source avec une architecture MVC (Model, View, Controller). Il permet aux développeurs de créer des applications web sans codage monotone et étendu. Grâce à ce framework, ils peuvent se servir d'outils pour réaliser un travail normé et organisé, poser les bases d'un projet web, suivre une architecture et une logique de travail strictes et rigoureuses. Cet applicatif est source de productivité pour un développeur car il dispose de "*composants*" qu'il est libre d'utiliser ou non dans son travail de développement web.

Twig

Twig est un moteur de templates pour le langage de programmation PHP, utilisé par défaut par le framework Symfony.

Doctrine

Doctrine est un ORM (Object Relational Mapping) qui permet de manipuler la base de données à travers des objets et non plus des valeurs brutes. Ceci permet une meilleure structuration du code et un lien fort entre l'utilisation et le stockage des données.

Copie écran du codage

```
<?php

namespace App\Entity;

use App\Repository\EvenementRepository;
use Doctrine\DBAL\Types\Types;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity(repositoryClass: EvenementRepository::class)]
class Evenement
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 50)]
    private ?string $titre = null;

    #[ORM\Column(length: 100)]
    private ?string $description = null;

    #[ORM\Column(type: Types::DATE_MUTABLE)]
    private ?\DateTimeInterface $date = null;

    #[ORM\Column(type: Types::TIME_MUTABLE)]
    private ?\DateTimeInterface $horaires = null;

    public function getId(): ?int
    {
        return $this->id;
    }

    public function getTitre(): ?string
    {
        return $this->titre;
    }

    public function setTitre(string $titre): self
    {
        $this->titre = $titre;
    }
}
```

```

        return $this;
    }

    public function getDescription(): ?string
    {
        return $this->description;
    }

    public function setDescription(string $description): self
    {
        $this->description = $description;

        return $this;
    }

    public function getDate(): ?\DateTimeInterface
    {
        return $this->date;
    }

    public function setDate(\DateTimeInterface $date): self
    {
        $this->date = $date;

        return $this;
    }

    public function getHoraires(): ?\DateTimeInterface
    {
        return $this->horaires;
    }

    public function setHoraires(\DateTimeInterface $horaires): self
    {
        $this->horaires = $horaires;

        return $this;
    }
}

```