**Programme sf1**

**Jour 1**

* Environnement travail :
  + Php 8
  + \*AMP
  + Vs Code (symfony, intelephense, namepsace resolver)
  + Composer
  + Symfony
* Installation projet Symfony
  + Webapp et mini
  + Symfony serve
  + Application opérationnelle
* Structure de dossier
  + Structure de dossier
  + Composer.json, .env / vendor / github
  + Modèle MVC
* POO
  + Symfony console debug :autowiring
  + Notion de POO
* Controller
  + Route (dd)
  + Response (dump)
  + AbstractController / render / vue.html
* Vue
  + Twig
  + Render(vue, param) / twig {{ }}
  + Make:controller
  + Extends (template)
  + Include (navbar)
* twig
  + {% path() %} / name-route
  + {% %} : for, if, set
  + Array : User.name / Objet : User.name (make:entity)
  + Filtre : lower, upper, date, length
* Request
  + Injection de Request
  + Dump($request)
  + $request->request / $request->query / session
* Entité
  + Création d’entité manuellement
  + Make:entity
  + Instanciation & renvoie vers la vue

**Jour 2**

* Doctrine
  + Qu’est-ce qu’un ORM ?
  + Entité représente les tables
  + Manager gère les lignes de la bdd (CUD)
  + Repository sélectionne des lignes dans la bdd
  + Doctrine et Bdd (.env, db :create, schema:update)
  + Migration
  + Fixtures
  + Afficher tous les posts
  + Afficher un post
* Category
  + Entité
  + fixtures
  + Afficher les category
  + Afficher les posts par category
  + Afficher par mot clef (DQL)

**Jour 3**

* Security authentication
  + Authentification (Qui es-tu ?)
  + Autorisation (As-tu le droit de faire ce que tu demandes à faire ?)
  + Firwall : zone définie par des urls ayant des politiques d’authentification différentes.
  + make:user (UserInterface) => security.yaml : password\_hasher, provider, firwall/main/provider
  + login
  + logout
  + make :registration-form
* Security autorisation
  + Ecrire du php (pas bonne pratiques)
    - $user = $security->getUser()

If(!inArray(‘ROLE\_ADMIN’, $user->getRoles())) {

return $this->redirectToRoute

}

* + Utiliser les raccourcies de l’abstract controller :
    - $this->getUser()
    - $this->isGranted(‘ROLE\_ADMIN’)
    - $this->denyUnlessIsGranted()
  + Utiliser les attributs sur une méthode ou un controller
    - #[isGranted(‘ROLE\_ADMIN’, status, message)
  + Security.yaml : access-control
  + $this->addFlash()

**Jour 4**

* Formulaire
  + Création du formulaire dans le contrôleur dans une classe dédiée (FormType)
  + Affichage du formulaire
  + Traitement des données
  + Validation des données

**Jour 5**

* Espace admin
* Faille csrf sur lien de suppression
* Formulaire EntityType & formulaire imbriqué