# Documentation générale Ishak’School

Table des matières

[Documentation générale Ishak’School 1](#_Toc201672547)

[Présentation du projet 2](#_Toc201672548)

[Objectifs pédagogiques 2](#_Toc201672549)

[Équipe 2](#_Toc201672550)

[Contexte 2](#_Toc201672551)

[Fonctionnalités principales 3](#_Toc201672552)

[Architecture technique 3](#_Toc201672553)

[Modules métiers 3](#_Toc201672554)

[Public visé 3](#_Toc201672555)

[Organisation projet 4](#_Toc201672556)

## Présentation du projet

Ishak’School est une plateforme web de gestion éducative conçue comme projet pédagogique. Elle permet aux utilisateurs de créer, lire, mettre à jour et supprimer (CRUD) des données liées aux élèves, enseignants, cours et inscriptions.

Ce projet met en œuvre une architecture moderne, modulaire et conteneurisée, en combinant :

* **React** pour le frontend dynamique et réactif,
* **Symfony API Platform** pour l’API back-end,
* **MariaDB** pour la base de données relationnelle,
* **PhpMyAdmin** pour la gestion visuelle de la BDD,
* **Nginx** en reverse proxy,
* **Docker** pour l’orchestration de tous ces composants.

## Objectifs pédagogiques

* Maîtriser le cycle complet d’un projet web fullstack.
* Mettre en place une API RESTful sécurisée et normalisée.
* Réaliser une interface frontend moderne, dynamique et responsive.
* Conteneuriser et déployer l’ensemble via Docker.
* Collaborer efficacement via GitHub et Jira.
* Expliquer les choix techniques dans un contexte projet réaliste.

## Équipe

|  |  |
| --- | --- |
| **Prénom** | **Rôle** |
| **Tümay YILDIZ** | Chef de projet & développeur frontend |
| **Ilias ASSAKDI** | Développeur frontend |
| **Yanis MERABET** | Développeur backend |
| **Fredy MENDES** | Développeur backend |
| **Elyas BELAÏDI** | Développeur backend |

## Contexte

Ishak’School est une école en pleine transition numérique. Elle souhaite centraliser les données de ses élèves, enseignants et cours dans un outil simple, sécurisé et accessible. L’application permet aux responsables de gérer les effectifs, les inscriptions et les cours, le tout via une interface intuitive et moderne.

## Fonctionnalités principales

* **Ajouter un élève**, un professeur ou un cours via un formulaire dynamique
* **Mettre à jour** les informations d’un utilisateur
* **Supprimer** des éléments en base (élève, prof, cours, inscription)
* **Lister** les entités existantes via une table interactive
* **Filtrer** les résultats (ex : élèves d’un cours, profs par matière)
* **Notifier** l’utilisateur avec des messages de succès ou d’erreur

## Architecture technique

Voir documentation d’architecture pour les détails, mais en résumé :

* 5 conteneurs : React, Symfony, Nginx, MariaDB, PhpMyAdmin
* Communication via un réseau Docker unique
* Requêtes HTTP entre React et Symfony
* API sécurisée via JWT
* Formulaire frontend dynamique généré à partir de la sélection CRUD

## Modules métiers

|  |  |  |
| --- | --- | --- |
| **Entité** | **Description** | **Champs clés** |
| **Student** | Élève inscrit | Prénom, Nom, Email, Date naissance |
| **Teacher** | Enseignant | Nom, Email |
| **Course** | Cours proposé | Titre, Description, Enseignant |
| **Enrollment** | Inscription d’un élève à un cours | Élève, Cours, Date |

## Public visé

Le projet s’adresse à toute structure éducative souhaitant :

* Gérer ses données internes simplement
* Disposer d’une interface web moderne
* Avoir une architecture déployable et modifiable facilement

## Organisation projet

* **Méthodologie** : Agile simplifiée (daily + tâches Jira)
* **Suivi** : Jira (tickets découpés par module)
* **Versioning** : GitHub, conventions Git Flow
* **Livrables** :
  + Application Dockerisée,
  + Base de données fonctionnelle avec fixtures,
  + Documentation générale, technique, achitecture et slides de présentation