

1. Introduction (30 sec)

Bonjour, je m'appelle Miyuki CHERBAL.

Avant cette formation, j'étais femme au foyer.

Avant de venir en France, j'étais photographe scolaire au Japon, et j'ai fait mes études en histoire de l'art là-bas.

J'ai choisi cette formation parce que j'avais envie de comprendre le domaine du web, de savoir comment fonctionne un site internet et comment le développer.

Aujourd'hui, je vais vous présenter mon projet : Tosho.

2. Présentation du projet (1 min)

Tosho veut dire "bibliothèque" ou "livre" en japonais.

C'est une application web que j'ai créée pour une association japonaise dont je fais partie. Cet association est dirigée par des parents bénévoles.

Elle propose des cours de japonais pour des enfants d'origine japonaise, et possède une petite bibliothèque.

Aujourd'hui, pour gérer les prêts de livres, nous utilisons une application web qui a été développée par un ancien parent bénévole il y a plusieurs années.

Cette application fonctionne, mais elle a plusieurs limites.

Par exemple, pour ajouter un nouveau livre dans le catalogue des livres, il faut contacter le développeur initial, parce qu'il n'y a pas d'accès administrateur.

L'interface est aussi très simple, sans CSS, donc pas très intuitive.

Comme je suis parent bibliothécaire de l'association, j'ai eu l'idée de refaire complètement l'application, pour améliorer la gestion, la rendre plus moderne et plus autonome.

3. Conception du projet (10 min)

Objectif et MVP

L'objectif principal de Tosho est de permettre aux parents bibliothécaires de gérer facilement les prêts et les retours de livres, et de disposer d'une interface administrateur.

Le MVP se concentre sur la fonctionnalité essentielle :

Enregistrer un prêt et un retour et également rechercher les prêts par livre ou par famille.

Ces fonctionnalités sont indispensables pour que l'application remplisse sa mission principale.

En parallèle, l'interface admin est un objectif important, car elle permet de gérer l'ensemble des livres, des familles et des bénévoles. Elle donne tout le sens à la modernisation de l'application et permet aux bénévoles de travailler de manière autonome.

Les utilisateurs

Il y a deux types d'utilisateurs :

- Les administrateurs, qui gèrent les familles, les livres, les inventaires et les bibliothécaires. Les personnes qui ne sont pas admin, ne peuvent pas créer de compte lui-même.
- Et les bibliothécaires, qui enregistrent les prêts et les retours, et participent à l'inventaire une fois par an.

Chaque rôle a accès à des fonctionnalités spécifiques, ce qui garantit que les actions importantes soient effectuées par les bonnes personnes.

Fonctionnalités principales

pour les bibliothécaires

Il y a deux fonctionnalités principales. Premierement, Prêts et retours de livres, avec un contrôle précis pour éviter les erreurs de double prêt.

Chaque livre a un code unique créé par l'association.

Avec ce code, les bénévoles peuvent rechercher un livre dans l'application, puis enregistrer un prêt ou un retour.

Il est aussi possible de chercher un livre par mot-clé, ou de chercher une famille par son nom pour lui ajouter un emprunt ou retirer un livre rendu.

Deuxième fonctionnalité pour les bibliothécaires c'est l'inventaire. Les livres de l'assocation sont rangés dans quatre endroits différents, lors d'une inventaire annuelle, les admins créer une session d'inventaire pour chaque lieu.

et les bibliothécaires vérifient la présence des livres et peuvent signaler des problèmes : livre abîmé, pas au bon endroit ou sans étiquette.

Pour les admin il y a 4 fonctionnalités importantes.

Premièrement, la gestion des livres. Les admin peuvent ajouter, consulter, modifier ou supprimer les livres.

deuxiemement, la gestion des familles. CRUD ici aussi.

troisièmement, la gestion des sessions d'inventaire. créer des sessions d'inventaire pour 4 sites en précisant la date, le lieu, et gérer l'ouverture de session en choisissant 'ouverte', 'à venir', 'fermé', et 'terminé'. Une fois que l'inventaire est ouverte, les admins peuvent suivre l'avancement, voir les livres vérifiés ou signalés, et ensuite mettre à jour l'état de l'inventaire une fois les problèmes réglés.

et quatrièmement la gestion des bibliothécaires. Seuls les admin peuvent créer des comptes bibliothécaires. une fois que l'admin créer un compte avec l'adresse mail de d'un bibliothécaire, un e-mail de notificatin qui dit que son compte a été crée avec un mot de passe. et le bibliothécaire va se connecter à son compte et changer le mot de passe.

Pour superviser la gestion de bibliothéque, les admin a également accès au interface bibliothécaire grâce à une bouton qui switch l'interface.

Organisation

Pour la conception, j'ai commencé par écrire toutes les fonctionnalités que je veux développer, et ensuite j'ai écrits un cahier des charges et userstories en markdown. Une fois que j'ai clarifié les besoins, j'ai créé des wireframes et maquettes sur Figma.

Charte graphique

J'ai choisi un design conviviale et ludique pour que les utilisateurs non techniques puisse se sentir familiale.

L'identité visuelle de Tosho a été pensée pour refléter l'esprit convivial de l'association et le côté ludique de l'école.

Les couleurs permettent de différencier facilement les interfaces selon le rôle de l'utilisateur.

Le site est entièrement responsive, pour que les bénévoles puissent l'utiliser facilement sur mobile comme sur desktop.

L'interface est intuitive, avec des icônes ludiques, afin de rendre la navigation simple et agréable pour tous.

Arborescence

Les principales pages :

- Accueil : choix entre "Admin" ou "Bibliothécaire"
- Page de prêt et retour

- Recherche de livre ou de famille
- Gestion des livres et utilisateurs

Stack technique

Tosho utilise :

- **Symfony 6.4** pour le back-end
- **Twig, CSS et JavaScript** pour le front
- **MySQL** pour gestion de base de données relationnel

MCD

Les principales tables :

- `book` : livres
- `family` : familles
- `loan` : emprunts
- `user` : admins et bibliothécaires

`inventory` : session de l'inventaire

`inventoryItem` : livre ajouté dans une session de l'inventaire

Chaque emprunt relie un **livre** et une **famille**.

4. Démo (6 à 10 min)

Je vais maintenant vous montrer Tosho.

Sur la page d'accueil, je choisis "bibliothécaire".

Je recherche une **famille**, puis un **livre**, et j'enregistre un **prêt**.

Quand le livre est rendu, je clique sur "retourner le livre".

Les informations se mettent à jour automatiquement.

Le site fonctionne aussi sur téléphone, avec la même clarté.

5. Front-end (5 min)

Pour faciliter la navigation, j'ai mis en place des onglets pour chaque page de fonctionnalité. par exemple, pour les prêts et les retours des livres, pour que l'accès aux champs de recherche soit

claire et pratique, j'ai mis un onglet pour recherche par famille, un autre onglet pour la recherche par livre.

Pour faciliter l'ajout de livres dans le catalogue, j'ai mis en place d'une fonctionnalité de préremplir le formulaire d'ajout d'un livre. L'utilisateur saisit un code ISBN, avec l'API, on récupère les infos sur les livres et préremplir le formulaire. J'ai utilisé deux API différents pour récupérer les données différentes. openlibrary pour récupérer les titres et l'auteur en romaji. romaji est un alphabet latin pour faciliter la lecture en japonais. Et l'URL de couverture. Deuxième API est openBd, c'est un API japonais, pour récupérer les titres et les auteurs en japonais. Cette fonctionnalité est développée avec framework JS Stimulus. Cette fonctionnalité a augmenté la qualité de l'expérience utilisateur.

- responsive avec des `media queries` est aussi pensé. Sur l'écran mobile, les composants sont adaptés correctement.

6. Back-end (5 min)

Pour le back, j'ai utilisé le modèle **MVC** de Symfony.

Exemple : quand on fait un prêt.

- Le contrôleur reçoit les données du formulaire.
- Il crée un nouvel objet **Loan**.
- Doctrine enregistre dans la base MySQL.
- La page affiche le prêt.

C'est un exemple simple de **CRUD** : créer, lire, modifier, supprimer.

7. Test (2-3 min)

8. Roadmap (1 min)

Pour la suite, j'aimerais ajouter :

- une vraie **connexion sécurisée** avec rôles
- un **historique des prêts** pour chaque famille
- une **recherche avancée**
- et peut-être un **QR code** pour enregistrer les livres plus vite.

9. Conclusion (3 min)

J'ai rencontré des difficultés, surtout avec les **relations entre tables** et le **contrôleur**.

J'ai trouvé des solutions grâce à la **documentation Symfony** et **Stack Overflow**.

Par exemple, pour un problème de jointure, j'ai trouvé une réponse en anglais et je l'ai adaptée à mon projet.

Ce projet m'a appris à **organiser mon travail**, à **structurer mon code**, et à **comprendre Symfony**.

Je suis très content d'avoir créé une application utile, pour une vraie école.

Tosho aide les bénévoles à gagner du temps et à mieux gérer la bibliothèque.

C'est un projet dont je suis fier.