

Excellent 👍

Voici une **version simplifiée** et fluide de ton **script oral** pour **Tosho**, avec des phrases courtes, naturelles, faciles à retenir et à dire à voix haute.

Tu peux la lire sans diapositives, ou t'en servir comme texte pour t'entraîner à parler.

Durée totale : environ **35 minutes avec pauses et transitions**.

Script oral simplifié – Projet Tosho

1. Introduction (30 sec)

Bonjour, je m'appelle [ton prénom].

Avant la formation, je faisais [ton ancienne activité].

J'ai choisi le développement web parce que j'aime créer des outils utiles, et comprendre comment les sites fonctionnent.

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 **école japonaise associative près de Lyon**.

Cette école est gérée par des **parents bénévoles**.

Elle enseigne le japonais à des enfants d'origine japonaise, et elle a une petite bibliothèque.

Mais avant Tosho, la gestion se faisait avec une vieille application :

pas d'accès admin, pas de recherche en japonais, et une interface difficile à utiliser.

J'ai donc voulu créer une **solution moderne, simple et claire**, adaptée aux bénévoles.

3. Conception du projet (10 min)

Objectif et MVP

Mon objectif : permettre aux bénévoles d'enregistrer les **prêts** et les **retours** de livres facilement.

Le MVP :

- Emprunter un livre
- Rendre un livre
- Rechercher une famille ou un livre

Les utilisateurs

Il y a deux types d'utilisateurs :

- **Administrateurs** : ils gèrent les familles, les bénévoles, les livres et les inventaires.
- **Bibliothécaires** : ils enregistrent les prêts et les retours.

Fonctionnalités principales

- Prêts et retours
- Inventaire et signalement de livres perdus ou abîmés
- Gestion des livres, familles et bénévoles (ajout, modification, suppression)

Organisation

J'ai travaillé en **sprints**, avec une méthode **Agile**.

J'ai utilisé **Trello** pour suivre mes tâches, et écrit mes **user stories** en Markdown.

Les priorités :

0 = obligatoire, 1 = important, 2 = secondaire.

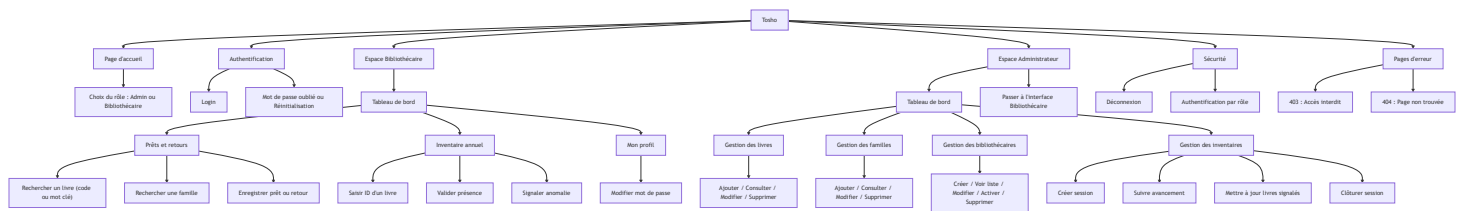
Charte graphique

J'ai choisi un design **clair et apaisant**, comme une vraie bibliothèque.

Les couleurs sont douces, et le site est **responsive**.

J'ai voulu une interface facile à utiliser, surtout sur mobile.

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

Wireframes

J’ai fait mes **maquettes** avant de coder.

J’ai ensuite gardé la même structure dans le HTML et le CSS.

Stack technique

Tosho utilise :

- **Symfony 6.4** pour le back-end
- **Twig, CSS** et **JavaScript** pour le front
- **MySQL** pour la base de données
- et **Doctrine ORM** pour gérer les relations entre les tables

MCD

Les principales tables :

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

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 le front, j'ai soigné la structure du code.

J'ai séparé mes fichiers Twig : `_search_family_tab` , `_search_book_tab` , `_loan_card` , etc.

Ça rend le code plus propre et plus facile à lire.

J'ai aussi ajouté :

- des effets au survol,
- des `required` dans les formulaires,
- et du responsive avec des `media queries` .

Les onglets de recherche suivent bien le design de ma maquette.

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)

Pour les tests, j'ai utilisé **Postman** et **phpMyAdmin**.

J'envoie une requête pour créer un emprunt.

Ensuite, je vérifie dans la base que les données sont bien enregistrées.

Ça montre que le front et le back communiquent bien.

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.

Souhaites-tu que je te fasse maintenant une **fiche de révision ultra courte** (par exemple 1 phrase par slide) pour t'aider à t'entraîner sans lire tout le texte ?

C'est pratique pour répéter à voix haute sans tes notes.