lements ajoutés en plus par rapport aux consignes :

* Un signe de panier associé à chaque utilisateur. Chaque utilisateur a un panier qui peut recevoir des livres, calculer des prix, gérer des quantités. Quand l’utilisateur se connecte, il retrouve son panier.
* Un système de privilèges. J’ai fait une table status, avec les champs status-name et status-state. Status-state 2 = Admin, 1 = user. La table user-status associe les ID des utilisateurs aux Status-State.
* Mis à part l’admin, chaque utilisateur est créé avec le statut user et n’a pas les privilèges de l’administrateur.
* J’ai ajouté une page mon compte
* Une page bibliothèque qui liste les livres par groupe de trois, avec possibilité de recherche par nom d’auteur et de catégories.

Lundi 29 septembre :

* J’ai fait quelques tests sur Symfony et Nuxt, histoire d’établir un premier squelette pour l’application.

Mardi 30 septembre :

* Configuration du repo github du projet.
* J’ai réellement commencé à travailler sur le projet Nuxt-Symfo à 9h00.
* 15h00 : J’avais terminé le système de logging ainsi que l’affichage des livres dans la page d’accueil
* 19h00 : J’ai finalisé de set le système de log pour les admins, et les menus spéciaux pour les administrateurs. On peut accéder au management des livres pour le header.
* J’ai créé des entités catégories, et auteur à part.
* J’ai fait une grosse refactorisation de la charte graphique.
* J’ai configuré l’API pour les users, status des utilisateurs (admins et users), livres, auteurs, catégories.
* J’ai fait un système de tokens pour préserver les sessions. Quand on se connecte, on stocke un token dans le local storage. Ce qui permet de préservation la session quand on rafraîchit la page depuis le front. Ce token est passé à la page default.vue pour être passé ensuite à l’ensemble des pages.
* J’ai refait le système de panier pour la commande de livre
* J’ai refait le système permettant la recherche de livres, par auteur, nom et catégories dans la page des livres.
* La spécificité du panier, est qu’on y accède par l’URL <http://localhost:8000/api/cart/current-user>
* Donc uniquement par l’utilisateur courant. On peut pas accéder au panier d’un autre utilisateur par un URL, ce qui est plus sécurisant.

Mercredi :

* J’ai apporté des corrections à l’API, et adapté plusieurs contrôleurs ainsi que les vues à l’API.
* J’ai créé le système de management de livres avec la vue book management, avec le système de création d’auteurs et de catégories.

Jeudi :

* Le fichier Book-Management faisait 1300 lignes de code. J’ai séparé la modifications de catégories, d’auteurs, de livres, et la création de ces entités dans des composants destinés à remplir ce rôle : AuthorCreationmodal.vue AuthorModificationModal.vue BookCreationModal.vue, ModifyBookModal.vue, CategoryCreateModal.vue, CategoryModifyModal.vue. C’est plus propre, et ça sépare les responsabilités, donc c’est plus facile à debug.
* Sécurisation de la page book-management.
* Petit cadeau personnel bonus : J’ai réalisé un mode responsive. On ne doit pas négliger les utilisateurs mobile. En 2025 ce serait une erreur. Bien sûr il est pas parfait. Mais ça peut faire office de demo.

Temps de travail estimé :

Lundi : 9h00

Mardi : +9h00 = 18h00

Mercredi +8h00 = 26h00

Jeudi +6h00 = +/- 32h00.

Total : 31h00