WEBAPP : Étape 2 - Réalisation du site

Auguste Celerier, Julie Descloitres, Tchadel Icard, Eric Khella, Jean-Philippe Levesques, Yiré Asma Soro

31 mars 2025

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Table des matières

1 Introduction

Le projet consiste à développer un site web permettant à un utilisateur de créer une liste de souhaits et d'offrir des cadeaux à partir des listes de souhaits d'autres utilisateurs. Le site devra inclure les cas d'utilisation suivants et être composé des pages suivantes :

1.1 Cas d'utilisation 1 : création de liste de souhaits

- Une page d'inscription et de connexion pour les utilisateurs. **(INSERT, READ)**
- Une page listant les listes de souhaits d'un utilisateur, avec les fonctionnalités suivantes :
  - Création d'une nouvelle liste (nom + date limite de réception des cadeaux),
  - Acceptation ou refus d'une invitation à une liste de souhaits,
  - Modification ou suppression d'une liste de souhaits existante,
  - Génération d'un lien URL pour partager une liste avec un autre utilisateur inscrit (création conjointe),
  - Génération d'un lien URL permettant à tous les utilisateurs d'accéder à la liste et d'acheter des cadeaux. **(INSERT, READ, UPDATE, DELETE)**
- Une page pour ajouter, modifier ou supprimer un article dans une liste de souhaits, incluant :
  - Un titre,
  - Une description de l'article,
  - Une URL menant à la page d'achat de l'article. **(INSERT, READ, UPDATE, DELETE)**

1.2 Cas d'utilisation 2 : achat d'un cadeau depuis une liste de souhaits

- Une page permettant de visualiser une liste de souhaits à partir de son lien de partage, avec les options de tri suivantes :
  - Trier par prix croissant,
  - Trier par prix décroissant.

  Chaque article possède un bouton d'achat qui redirige vers la page de vente et ouvre une autre page pour télécharger une preuve d'achat. Lorsqu'un article est déjà acheté, le nom de l'acheteur et un message de félicitations apparaissent. **(READ)**
- Une page permettant de prouver l'achat d'un article en téléchargeant une capture d'écran de la preuve d'achat et en ajoutant un message de félicitations. L'acheteur pourra ultérieurement modifier ce message. **(INSERT, READ, UPDATE)**

1.3 Cas d'utilisation 3 : administration

- Un tableau de bord affichant :
  - Le classement des trois articles les plus chers achetés pour une liste de souhaits,
  - Le classement des trois listes de souhaits ayant reçu les cadeaux les plus onéreux. **(READ)**
- Une page permettant de consulter la liste des utilisateurs et de gérer leurs comptes (verrouillage/déverrouillage/suppression). **(READ, UPDATE, DELETE)**

1.4 Accès au code source

Le code source du projet ainsi que l'ensemble de la documentation sont disponibles sur le dépôt Git à l'adresse suivante : https://gitlab-df.imt-atlantique.fr/t22icard/webapp-equipe7.

2 Répartition des tâches entre les binômes

2.1 Binôme 21 : Yiré Asma Soro et Tchadel Icard

Cas d'utilisation 1 : Création de liste de souhaits

- Page d'inscription et de connexion pour les utilisateurs
  Réalisée par : **Tchadel Icard**
- Page listant les listes de souhaits d'un utilisateur, avec les fonctionnalités suivantes :
  - Création d'une nouvelle liste (nom + date limite de réception des cadeaux)
  - Acceptation ou refus d'une invitation à une liste de souhaits
  - Modification ou suppression d'une liste de souhaits existante

  Réalisée par : **Yiré Asma Soro**
- Génération d'un lien URL permettant à tous les utilisateurs d'accéder à la liste et d'acheter des cadeaux
  Réalisée par : **Tchadel Icard**
- Page pour ajouter, modifier ou supprimer un article dans une liste de souhaits, incluant :
  - Un titre
  - Une description de l'article
  - Une URL menant à la page d'achat de l'article

  Réalisée par : **Tchadel Icard**

2.2 Binôme 19 : Jean-Philippe Levesques et Julie Descloitres

Cas d'utilisation 2 : Achat d'un cadeau depuis une liste de souhaits

- Une page permettant de visualiser une liste de souhaits à partir de son lien de partage, avec les options de tri suivantes :
  - Trier par prix croissant,
  - Trier par prix décroissant.

  Chaque article possède un bouton d'achat qui redirige vers la page de vente et ouvre une autre page pour télécharger une preuve d'achat. Lorsqu'un article est déjà acheté, le nom de l'acheteur et un message de félicitations apparaissent.
  Réalisée par : **Jean-Philippe Levesques**
- Une page permettant de prouver l'achat d'un article en téléchargeant une capture d'écran de la preuve d'achat et en ajoutant un message de félicitations. L'acheteur pourra ultérieurement modifier ce message.
  Réalisée par : **Julie Descloitres**

2.3 Binôme 20 : Auguste Celerier et Eric Khella

Cas d'utilisation 3 : Administration

- Un tableau de bord affichant :
  - Le classement des trois articles les plus chers achetés pour une liste de souhaits.
    Réalisé par : **Eric Khella**

  - Le classement des trois listes de souhaits ayant reçu les cadeaux les plus onéreux.
    Réalisé par : **Auguste Celerier**
- Une page permettant de consulter la liste des utilisateurs et de gérer leurs comptes (verrouillage/déverrouillage/suppression).
  Réalisé par : **Eric Khella**

3 Choix technologiques

Pour la réalisation de ce projet, nous avons choisi d'utiliser le framework Symfony plutôt que de développer l'application en pur PHP sans framework. Ce choix repose sur plusieurs raisons essentielles qui ont facilité le développement et garanti la robustesse du projet.

3.1 Gain de temps et simplicité de développement

Symfony est un framework complet et structuré qui offre une architecture modulaire et claire. Cela permet de développer des applications web de manière rapide et efficace, en réduisant le temps nécessaire à la mise en place des fonctionnalités de base telles que l'authentification, la gestion des routes ou encore la gestion des formulaires.

Grâce aux outils intégrés et aux bonnes pratiques encouragées par Symfony, nous avons pu nous concentrer sur le développement des fonctionnalités propres au projet sans réinventer des composants déjà existants et éprouvés. En particulier, l'utilisation des composants Symfony tels que les formulaires, les routes et la gestion des services a considérablement simplifié l'écriture du code.

3.2 Sécurité renforcée

L'un des principaux avantages de Symfony est la gestion native des aspects de sécurité. Le framework offre une protection contre les failles courantes, notamment :

- **Protection contre les attaques CSRF (Cross-Site Request Forgery) :** Symfony fournit des tokens CSRF pour sécuriser les formulaires et empêcher les soumissions non autorisées.
- **Prévention des failles XSS (Cross-Site Scripting) :** L'utilisation des templates Twig garantit une échappement des caractères spéciaux pour éviter l'injection de scripts malveillants.
- **Protection contre les injections SQL :** En utilisant l'ORM Doctrine, les requêtes sont automatiquement paramétrées, évitant ainsi les risques d'injections SQL.

Ces fonctionnalités permettent de renforcer la robustesse et la sécurité de l'application sans avoir à implémenter manuellement des mécanismes complexes et susceptibles d'erreurs.

3.3 Gestion des données avec Doctrine

Symfony intègre nativement l'ORM Doctrine, qui facilite la gestion de la base de données en offrant une abstraction de la couche SQL. Plutôt que d'écrire manuellement des requêtes complexes, nous avons pu manipuler les données via des objets en utilisant des entités.

Doctrine propose également un système de migrations pour suivre les évolutions du schéma de la base de données, ce qui est essentiel pour maintenir la cohérence des données lors des mises à jour de l'application.

3.4 Écosystème riche et maintenu

Symfony bénéficie d'une vaste communauté et d'un écosystème mature, avec de nombreux bundles permettant d'ajouter facilement des fonctionnalités avancées. Cette richesse a permis de trouver rapidement des solutions aux problématiques rencontrées et de tirer parti des bonnes pratiques de développement.

4 Points forts et points faibles du projet

4.1 Points forts

Le projet présente plusieurs points forts notables qui démontrent la réussite de notre démarche de développement malgré les contraintes de temps et d'organisation.

- **Fonctionnalités complètes :** Nous avons implémenté l'ensemble des fonctionnalités spécifiées dans le cahier des charges du projet. Cela inclut la création et la gestion des listes de souhaits, l'ajout d'articles, l'achat de cadeaux avec preuve d'achat, et la gestion des utilisateurs avec un espace d'administration. Chaque cas d'utilisation a été pris en compte et réalisé conformément aux spécifications.
- **Fonctionnalités supplémentaires :** Au-delà des exigences initiales, nous avons intégré des notifications par mail pour améliorer l'expérience utilisateur. Ainsi, chaque utilisateur reçoit un email lui demandant de valider son compte lors de sa création, et des invitations par mail sont envoyées lorsqu'un utilisateur est ajouté à une wishlist. Ces ajouts améliorent la sécurité et l'expérience utilisateur.
- **Respect de la charte graphique :** Nous avons pris soin de respecter la charte graphique d'IMT Atlantique afin de garantir une cohérence visuelle et une identité graphique harmonieuse. Bien que certains détails restent perfectibles, l'ensemble du site présente une bonne lisibilité et une esthétique soignée.
- **Responsivité :** Le site est globalement responsive, ce qui le rend utilisable sur différents appareils, y compris les ordinateurs, les tablettes et les smartphones. Certaines pages ou éléments peuvent cependant nécessiter des ajustements pour une meilleure adaptabilité.
- **Bonne structure de projet :** Grâce au framework Symfony, nous avons structuré notre code de manière modulaire et organisée. Les composants de base, les entités, les contrôleurs et les vues sont bien séparés, ce qui permet une maintenance plus facile et évolutive. De plus, l'usage de l'ORM Doctrine garantit une gestion des données efficace et sécurisée.
- **Sécurité renforcée :** l'utilisation des fonctionnalités de sécurité de Symfony, telles que les tokens CSRF et les protections contre les failles XSS et SQL injection, assure un niveau de protection élevé contre les attaques potentielles.

4.2 Points faibles

Malgré ces points forts, nous avons rencontré certaines difficultés et limitations dans la réalisation du projet, principalement dues aux contraintes de temps et à la collaboration en parallèle des équipes.

- **Propreté du code :** Le projet ayant été réalisé en une semaine, la qualité du code n'est pas optimale. En particulier, nous avons concentré la logique métier dans les contrôleurs au lieu de la répartir dans des services dédiés. Cela rend certaines parties du code difficiles à maintenir et à réutiliser.
- **Optimisation des requêtes :** Certaines opérations auraient pu être optimisées en utilisant des requêtes avancées directement dans les repositories. Par exemple, des jointures plus complexes ou des agrégations auraient permis de réduire la charge sur le serveur d'application en déléguant davantage de traitement à la base de données.
- **Séparation du style :** Compte tenu du travail collaboratif en parallèle, nous n'avons pas tou-

jours respecté une séparation stricte entre le code HTML et les fichiers CSS. Il est fréquent que des styles soient définis directement dans les balises HTML, ce qui nuit à la maintenabilité et à la lisibilité du code. Une centralisation des styles dans des fichiers CSS aurait permis de rendre le projet plus homogène et plus facile à faire évoluer.

— **Vérification des fichiers téléchargés :** La gestion des preuves d'achat permet aux utilisateurs d'envoyer des captures d'écran. Cependant, nous n'avons pas complètement sécurisé la vérification des types de fichiers uploadés. Cela pourrait entraîner des risques de sécurité ou des comportements inattendus si des fichiers non conformes sont téléchargés.

Bien que ces points faibles aient un impact sur la qualité globale du projet, nous avons toutefois su surmonter les principaux défis pour proposer une application fonctionnelle et conforme aux exigences.

5 Difficultés rencontrées par chaque binôme lors du développement du projet

5.1 Binôme 19 : Jean-Philippe Levesques et Julie Descloitres

5.1.1 Gestion du *use-case* sans lien de partage

Pour commencer, la première difficulté rencontrée a été la création du Use-case sans avoir le lien de partage sur lequel se concentrer. En effet, chaque binôme devait avancer en parallèle et nous devions créer notre section sans avoir l'URL de partage, pas encore aboutie pour développer notre page. Nous avons donc commencé par créer une base de données fictive afin de tester au fur et à mesure de la construction de la page la validité de notre code. Cette technique a bien marché et nous a permis, une fois que notre code était correct, de supprimer cette liste fictive de produits que nous avions créée en direct sur le `WishlistController` afin de la remplacer par le lien de partage que le binôme 21 a pu créer et réaliser quelques ajustements pour que nos 2 tâches fonctionnent.

5.1.2 Ajout de la fonction tri

Par la suite, l'une des principales difficultés rencontrées a été l'implémentation de la fonction de tri des articles par prix. Il a fallu manipuler le DOM en JavaScript pour récupérer les prix des articles, les convertir en nombres exploitables, puis réorganiser dynamiquement les éléments de la liste en fonction de l'ordre choisi (croissant ou décroissant). Cette tâche a demandé une bonne compréhension de la manipulation des nœuds HTML et de l'impact des modifications dynamiques sur l'affichage des données. Nous avons pu nous aider de Stack Overflow pour comprendre le fonctionnement du DOM et réaliser correctement la fonction `sortByPrice`. De plus, il a fallu s'assurer que le tri restait fonctionnel et performant même après plusieurs interactions.

5.1.3 L'ouverture de deux pages simultanément

Un autre défi technique a été de permettre l'ouverture simultanée de deux pages lors du clic sur le bouton « Acheter » : l'une redirigeant vers la page d'achat et l'autre vers la page de dépôt de preuve d'achat. L'enjeu ici était de trouver un moyen efficace d'ouvrir ces deux liens sans être bloqué par les restrictions des navigateurs, qui peuvent empêcher l'ouverture multiple de fenêtres. L'utilisation de `window.open()` a permis de contourner cette contrainte, mais il a fallu tester différents paramètres et méthodes afin de garantir que les deux pages s'ouvrent bien ensemble sans interférer l'une avec l'autre. Nous avons donc opter pour la création d'une fonction pour la réalisation de cette tâche.

5.1.4 Création des pages purchase

La difficulté principale pour la création de la page de *purchase* a été de gérer le téléversement de la preuve, et sa destination. En effet, les noms des fichiers téléchargés semblaient poser problème. Nous avons réussi à régler le problème en suivant un exemple trouvé dans la documentation utilisant un *slugger*. Celui-ci permet de récupérer une chaîne de caractères à partir du nom du fichier qui soit facile à utiliser (e.g. sans caractères spéciaux, ...).

Aussi, nous avons rencontré des difficultés pour la partie *update* de la page *purchase*. Nous avons initialement essayé d'utiliser la même méthode pour la création d'une *purchase* et la mise à jour de celle-ci, mais comme nous avons exigé la présence d'une pièce justificative dans le formulaire de création, la modification simple du message ne fonctionnait pas. Nous avons donc décidé de faire un formulaire séparé pour la mise à jour du message de félicitations et/ou de la preuve d'achat, en enlevant cette fois-ci l'exigence de la preuve d'achat afin de pouvoir modifier l'un, l'autre ou les deux.

5.2 Binôme 20 : Auguste Celerier et Eric Khella

5.2.1 Fonctionnalité de suppression d'un utilisateur

Nous avons rencontré un problème dans le système de suppression des utilisateurs. Le bouton de suppression sur l'interface fonctionnait en apparence, mais ne supprimait pas réellement les utilisateurs de la base de données : seule l'interface graphique se mettait à jour temporairement. Il a fallu revoir le contrôleur responsable de cette action, s'assurer que le bon formulaire était utilisé avec le bon token CSRF, et que l'appel à Doctrine était correct.

5.2.2 Gestion des utilisateurs bloqués

Nous avons mis en place une fonctionnalité permettant à un administrateur de bloquer un utilisateur. Cependant, lorsqu'un utilisateur bloqué tentait de se connecter, le système renvoyait un message d'erreur générique de type « Bad credentials », ce qui pouvait prêter à confusion. Nous avions pourtant implémenté une exception personnalisée afin d'indiquer explicitement que le compte était bloqué. Après investigation, nous avons compris que cette exception était bien levée, mais interceptée en amont par la couche de sécurité de Symfony, qui renvoyait une erreur par défaut. Pour corriger cela, nous avons dû configurer un `AuthenticationFailureHandler` personnalisé afin d'intercepter les erreurs de connexion et d'afficher un message plus explicite à l'utilisateur bloqué.

5.2.3 Difficultés de coordination

En parallèle des problèmes techniques, nous avons aussi rencontré des difficultés d'ordre organisationnel. Il nous est arrivé de modifier simultanément les mêmes fichiers ou services, ce qui a entraîné des conflits git ou des effets de bord imprévus. Pour pallier cela, nous avons progressivement mis en place une organisation plus claire, avec une répartition explicite des rôles et une meilleure communication via messages réguliers et vérification mutuelle des modifications.

5.3 Binôme 21 : Yiré Asma Soro et Tchadel Icard

Lors du développement du projet, notre binôme a rencontré plusieurs difficultés techniques et organisationnelles que nous avons dû surmonter.

5.3.1 Gestion des invitations dans les wishlists

La première difficulté majeure concernait la mise en place du système d'invitation dans les wishlists. Nous avions initialement envisagé un système où l'utilisateur pourrait saisir les adresses e-mail des

collaborateurs. Ce système aurait envoyé automatiquement un e-mail contenant un lien d'invitation permettant de rejoindre la wishlist après connexion. Cependant, en raison de contraintes de temps, cette approche s'est avérée trop complexe à mettre en place.

Pour simplifier l'implémentation, nous avons opté pour une solution plus directe consistant à proposer une liste déroulante avec l'ensemble des utilisateurs de l'application. L'utilisateur peut alors cocher les personnes qu'il souhaite inviter dans la wishlist. Bien que cette solution soit moins flexible, elle a permis d'implémenter la fonctionnalité dans le temps imparti tout en restant fonctionnelle.

5.3.2 Gestion des conflits de merge

Au début du développement, nous avons également rencontré des problèmes liés aux conflits de merge. En effet, travaillant en parallèle sur des parties communes, notamment lors de la création des pages de base du site, nous avons souvent généré des conflits dans le code source. La coordination et la gestion des versions ont posé problème dans un premier temps.

Pour surmonter cette difficulté, nous avons amélioré notre organisation en nous attribuant des parties plus distinctes et en renforçant la communication au sein du binôme. Finalement, nous avons réussi à aboutir à un site cohérent avec des pages au style harmonisé, malgré les défis de collaboration initiale.

5.3.3 Problème d'accessibilité des pages publiques

Une autre difficulté importante que nous avons rencontrée concernait la gestion des pages publiquement accessibles avec le bundle de sécurité de Symfony. La documentation officielle ayant évolué au fil des versions, nous avons eu des difficultés à rendre certaines pages accessibles sans authentification. En particulier, nous utilisions initialement l'ancien rôle `IS_AUTHENTICATED_ANONYMOUSLY`, alors que la version de Symfony utilisée requérait le rôle `PUBLIC_ACCESS`.

Ce changement subtil mais crucial nous a fait perdre beaucoup de temps en phase de débogage, car les informations disponibles sur internet étaient parfois contradictoires. Après plusieurs essais et consultations des sources communautaires, nous avons finalement réussi à corriger ce problème et à rendre les pages correctement accessibles.

Malgré ces difficultés, nous avons pu finaliser le projet en surmontant les principaux obstacles techniques et en améliorant progressivement notre organisation.

6 Conclusion

La réalisation de ce projet a représenté un véritable défi, tant sur le plan technique qu'organisationnel. En une semaine seulement, nous avons réussi à développer un site web complet et fonctionnel, répondant aux exigences définies dans le cahier des charges. Grâce à l'utilisation du framework Symfony, nous avons pu exploiter les bonnes pratiques de développement tout en bénéficiant de fonctionnalités avancées pour la gestion de la sécurité et des données.

Ce projet a également constitué un excellent exercice d'apprentissage dans le cadre de l'UE **WEBAPP**. Il nous a permis de mettre en pratique nos connaissances en développement web tout en découvrant des aspects plus complexes de l'architecture d'une application moderne. Nous avons pu approfondir notre maîtrise de Symfony, de la gestion des données avec Doctrine, ainsi que des mécanismes de sécurité avancés fournis par le framework.

Malgré les contraintes de temps et les difficultés rencontrées, notamment dans la gestion des invitations, les conflits de merge et les problèmes de configuration de la sécurité, nous avons su faire preuve d'adaptabilité et d'efficacité. Les décisions prises, même lorsqu'elles impliquaient des compromis, ont permis d'assurer un niveau de qualité satisfaisant et de livrer un produit robuste.

L'implication de chaque membre de l'équipe a été essentielle pour atteindre ce résultat. La communication et la coordination ont parfois été un défi, mais nous avons su nous organiser pour garantir l'avancement du projet.

Pour conclure, ce projet nous a permis de développer des compétences concrètes en développement web avec Symfony et de travailler efficacement en équipe. Les difficultés rencontrées nous ont permis d'apprendre et de progresser, et le résultat final reflète les efforts de tout le groupe. Même si le site reste perfectible, il représente un résultat satisfaisant et montre notre capacité à faire face aux défis malgré les contraintes de temps.