

# Développement Full-Stack - Projet 2025

## Plateforme de Réseau Social Web

### Présentation Générale du Projet

Le thème retenu pour ce projet est la conception et la réalisation d'une plateforme de réseau social web, inspirée des plateformes de discussion modernes, intégrant des notions avancées de gestion des utilisateurs, d'interactions sociales et de scalabilité.

Le projet consiste à développer une application web de réseau social reposant sur un modèle de discussions par threads. La plateforme doit permettre aux utilisateurs de créer du contenu, d'interagir entre eux, de gérer leur visibilité et de recevoir des notifications.

L'application doit être conçue comme un produit professionnel, prêt pour un contexte de production. Elle devra s'appuyer sur une architecture full stack moderne, incluant un back-end exposant une API REST sécurisée, un front-end sous forme d'application monopage (SPA) et une base de données robuste.

### Objectifs Pédagogiques et Compétences Visées

Ce projet a pour objectif de valider l'acquisition des compétences clés d'un développeur web full stack junior. À travers ce projet, l'étudiant doit démontrer sa capacité à comprendre un besoin fonctionnel complexe, à le traduire en une architecture technique cohérente et à implémenter une solution fiable et maintenable.

Les compétences visées incluent la maîtrise du développement web front-end et back-end, la conception de bases de données relationnelles ou non relationnelles, la sécurisation des applications, la gestion des états et des données côté client, ainsi que l'utilisation d'outils professionnels de versioning, de déploiement et de gestion de projet.

Le projet vise également à développer des compétences transverses telles que l'analyse, la modélisation UML, la planification, la documentation technique et l'adoption d'une vision orientée produit et start-up.

### Périmètre Fonctionnel de la Plateforme

#### Authentification et Gestion des Comptes

La plateforme devra intégrer un système d'authentification complet permettant aux utilisateurs de s'inscrire, de se connecter et de gérer leurs sessions de manière sécurisée. Les mécanismes d'authentification devront respecter les standards du marché, notamment en matière de hachage des mots de passe et de gestion des jetons d'accès.

Les utilisateurs devront pouvoir choisir entre un compte public ou privé. Les comptes publics sont accessibles à tous les utilisateurs, tandis que les comptes privés nécessitent une validation des demandes de suivi.

## **Profils Utilisateurs**

Chaque utilisateur disposera d'un profil personnel présentant ses informations publiques, une photo de profil, une biographie et des indicateurs d'activité. Le profil constitue l'identité numérique de l'utilisateur sur la plateforme et doit respecter les règles de confidentialité définies dans les paramètres du compte.

Les utilisateurs devront pouvoir modifier leurs informations personnelles via une interface sécurisée.

## **Threads et Publications**

Le cœur fonctionnel de la plateforme repose sur le concept de threads. Les utilisateurs peuvent créer des threads contenant du texte et éventuellement des médias. Ces threads permettent d'initier des discussions et constituent l'unité principale de contenu de la plateforme.

Les threads doivent permettre des interactions telles que les réponses, les réactions et l'affichage de statistiques d'engagement. La conception devra prendre en compte la modération, la validation des contenus et les performances.

## **Fil d'Actualité**

Le fil d'actualité représente l'espace central de consultation des contenus. Il agrège les threads publiés par les utilisateurs suivis, ainsi que des contenus publics pertinents. Le fil doit être généré dynamiquement en fonction des relations sociales et des paramètres de confidentialité.

Des mécanismes de pagination ou de défilement infini devront être mis en place afin d'assurer une bonne expérience utilisateur.

## **Système de Suivi (Followers / Following)**

La plateforme doit permettre aux utilisateurs de suivre ou de ne plus suivre d'autres utilisateurs. Ce système influence directement le fil d'actualité et les notifications.

Dans le cas des comptes privés, les demandes de suivi doivent être soumises à validation. La cohérence et la performance de la gestion des relations sociales doivent être garanties.

## **Notifications**

Un système de notifications doit informer les utilisateurs des événements importants tels que les nouvelles demandes de suivi, les validations, les réponses aux threads ou les nouvelles interactions.

Les notifications seront initialement intégrées dans l'application et conçues de manière à pouvoir évoluer vers un système temps réel.

## Gestion des Médias

La plateforme doit permettre le téléchargement et la gestion de médias, notamment pour les photos de profil et les contenus des threads. Les fichiers doivent être validés, sécurisés et stockés de manière appropriée, localement ou via un service cloud.

## Paramètres et Préférences

Un module de paramètres doit permettre aux utilisateurs de gérer la confidentialité de leur compte, leurs préférences de notifications et leurs informations personnelles. Ces paramètres doivent être accessibles uniquement aux utilisateurs authentifiés.

## Architecture Technique

### Architecture Front-End

Le front-end devra être développé sous forme d'application monopage (SPA) avec **React.js**. L'architecture devra être basée sur des composants réutilisables, avec une gestion claire des états et des effets.

L'interface utilisateur devra être responsive et adaptée à différents supports, en s'appuyant sur des frameworks CSS tels que Bootstrap ou Tailwind CSS. Les échanges avec le back-end se feront via des appels HTTP sécurisés.

### Architecture Back-End

Le back-end devra être développé avec **Node.js** et **Express**, en respectant une architecture en couches incluant routes, contrôleurs, services et accès aux données. L'API devra exposer des endpoints REST permettant la gestion complète des fonctionnalités de la plateforme.

Des mécanismes de sécurité tels que la configuration CORS, la protection des en-têtes HTTP et la validation des données devront être mis en place.

### Conception de la Base de Données

La base de données pourra être relationnelle ou non relationnelle, selon le choix de l'étudiant. Le modèle de données devra représenter les utilisateurs, les threads, les relations sociales, les notifications et les médias.

L'utilisation d'un ORM ou d'un ODM est recommandée afin de structurer l'accès aux données et d'assurer la maintenabilité.

## DevOps, Déploiement et Qualité

L'application devra être conteneurisée à l'aide de **Docker**. Une approche CI/CD devra être définie afin d'automatiser les phases de build, de test et de déploiement.

Le code source devra être versionné avec **Git** et hébergé sur **GitHub**, en respectant des bonnes pratiques professionnelles. Des outils de gestion de projet tels que Trello ou Jira devront être utilisés pour suivre l'avancement.

## Conception, Modélisation et Documentation

L'étudiant devra fournir des diagrammes UML pertinents, incluant des diagrammes de cas d'utilisation, des diagrammes de classes ou de données, et des diagrammes de séquence.

Un diagramme de Gantt devra présenter les différentes phases du projet. Une documentation technique claire devra accompagner le projet afin d'expliquer les choix architecturaux, les technologies utilisées et le processus de déploiement.

## Conclusion

Ce projet de fin de formation a pour vocation d'évaluer la maturité technique et professionnelle de l'étudiant. La réalisation d'une plateforme de réseau social complète permet de démontrer la capacité à concevoir, développer et livrer une application web moderne répondant aux standards académiques et aux attentes du monde professionnel.