



Licence Professionnelle en Génie Logiciel

RAPPORT DE RÉALISATION DU PROJET TUTEURÉ 2

Titre du projet:

Thesis Manager,
la plateforme de gestion des soutenances

Intitulé de l'Unité d'enseignement : Projet Tuteuré 2

Code de l'Unité d'Enseignement : INF1525

Chargé du cours : M. AKAKPO

Tuteur : M. AKAKPO

Liste des membres du groupe :

Noms	Prénoms	Spécialité
ADAMEY	Kokou Daniel	GL
AKAKPOVI	Marie-Emmanuel C.A.	GL
ETEKPO	Afi Shina	GL

Sommaire

- 1. Introduction**
- 2. Calendrier de réalisation**
 - 2.1. Description**
 - 2.2. Planning**
- 3. Liste des livrables**
 - 3.1. Description**
 - 3.2. Liste et état actuel des livrables**
 - 3.3. Note sur la stabilité de la BD**
 - 3.4. Guide de déploiement et d'utilisation**
 - 3.4.1. Guide de déploiement**
 - 3.4.2. Guide d'utilisation**
- 4. Rapport de rencontres avec le professeur**
- 5. Conclusion**

1- Introduction

Le projet **Thesis Manager** s'inscrit dans une démarche de modernisation et de digitalisation des processus académiques de l'**Ecole Polytechnique de Lomé (EPL)**. Actuellement, la gestion des soutenances repose sur des méthodes manuelles (fichiers Excel, documents papier), ce qui engendre des erreurs de planification et une coordination difficile entre les acteurs.

L'objectif principal de ce projet est de concevoir et réaliser une plateforme web centralisée permettant d'automatiser la planification, de sécuriser le dépôt des mémoires et d'offrir une traçabilité totale du processus de soutenance. Ce rapport détaille la mise en œuvre technique des fonctionnalités clés, notamment le système d'authentification sécurisé par OTP, ainsi que les livrables produits.

2- Calendrier de Réalisation

2-1- Description

Pour parler du calendrier de réalisation, il faut le considérer comme la feuille de route opérationnelle du projet.

C'est un outil de gestion de projet qui décline le plan d'action théorique en une séquence temporelle concrète et visuelle.

Il ne se contente pas de fixer une date de début et de fin ; il détaille l'ensemble des tâches, des jalons clés, des livrables et des dépendances entre les différentes phases du projet.

Un bon calendrier de réalisation permet d'anticiper les goulots d'étranglement, d'allouer efficacement les ressources (humaines, financières, matérielles), et de suivre l'avancement par rapport aux prévisions.

Que ce soit sous forme de diagramme de Gantt, de tableau ou intégré à un logiciel de gestion, son objectif principal est de transformer une vision en actions planifiées, en répondant aux questions cruciales : "*Qui fait quoi ?*" et "*Quand cela doit-il être fait ?*".

2-2-Planning de réalisation

Activités	Descriptions	Durées	Dates de début de réalisation	Dates de fin de réalisation	Durées Effectives
Analyse des Besoins	Contexte, objectifs, acteurs, besoins, priorité, contraintes.	6jours	16-12-2025	21-12-2025	6jours
Conception du système	L'architecture de logiciel, schéma de base de données conceptions des maquettes diagramme techniques et choix des technologies	2jours	22-12-2025	23-12-2025	2jours
Développement et Tests	Mise en place d'une base de données développement des composants backend développement de l'interface frontend écriture exécution des tests unitaires réalisation des	1mois	05-01-2026	05-02-2026	1mois

	tests d'intégrations effectuer des tests fonctionnels corriger les anomalies				
Vérifications	Vérifications de la validité du codes, vérifications des performances, expérience utilisateurs	10jours	06-02-2026	16-02-2026	5jours

3- Liste des livrables

3.1. Description technique

La réalisation technique du projet **Thesis Manager** s'appuie sur une architecture moderne et modulaire, conçue pour répondre aux exigences de performance, de scalabilité et de sécurité définies dans le cahier des charges. Le choix des technologies a été dicté par la nécessité de centraliser et de fluidifier les interactions entre les trois profils d'utilisateurs : Étudiants, Enseignants et Administrateurs.

- **Frontend avec Next.js et Tamagui** : Pour l'interface utilisateur, nous avons opté pour le framework **Next.js**, qui permet un rendu hybride (statique et serveur) optimal pour les tableaux de bord de supervision et les portails de soumission. L'intégration de **Tamagui** garantit une interface réactive et une expérience utilisateur cohérente, que ce soit sur ordinateur ou sur support mobile.
- **Gestion des données via GraphQL et Relay** : Contrairement à une architecture REST classique, nous utilisons une **API GraphQL** pour la gestion des données. Ce choix permet de ne récupérer que les informations strictement nécessaires (par exemple, charger uniquement le titre et le statut d'un mémoire sans charger le fichier complet), réduisant ainsi la consommation de bande passante. Côté client, la bibliothèque **Relay** assure une gestion robuste et prévisible des données et des mutations (mises à jour) en temps réel.
- **Backend et Persistance** : Le cœur du système est piloté par une logique serveur structurée, s'appuyant sur une base de données relationnelle stable (PostgreSQL/MySQL) capable de gérer l'intégrité des relations entre les utilisateurs, leurs soumissions de documents et les sessions de soutenance.

- **Industrialisation et Qualité :** Le projet intègre des outils de développement modernes comme **ESLint** pour la qualité du code, ainsi que des scripts d'automatisation (`setup-submit-travail.sh`) pour faciliter le déploiement et la maintenance de la plateforme.

Cette stack technologique permet de transformer le processus de soutenance, autrefois manuel et fragmenté, en une solution numérique robuste, fiable et facile à prendre en main pour l'Université de Lomé.

3.2. Liste et état actuel des livrables

Voici l'état actuel des livrables :

Livrables	Description	État
Application Web (Frontend)	Interface développée avec Next.js et Tamagui pour un rendu multi-plateforme.	En cours
API GraphQL	Schéma de données défini (<code>schema.graphql</code>) pour une communication optimisée.	Stable
Base de Données (BD)	Structure relationnelle stable intégrant les tables Utilisateurs, Étudiants et Soumissions.	Stable
Logique Client (Relay)	Intégration de Relay pour la gestion performante des données et des mutations.	Stable
Documentation Technique	Guide de soumission des travaux (<code>SUBMIT_TRAVAIL_GUIDE.md</code>) et configuration ESLint.	Terminé
Scripts d'automatisation	Scripts de configuration et de déploiement (<code>setup-submit-travail.sh</code>).	Prêt
Système d'Authentification	Module de connexion intégré à l'infrastructure globale du projet.	En cours

3.3. Note sur la stabilité de la BD

La base de données est **stable**. Elle a été conçue pour supporter le workflow complet : du dépôt de mémoire par l'étudiant à la planification par l'administrateur, en respectant les tables essentielles (Utilisateur, Étudiant, Enseignant, Soumissions, Soutenance) définies dans le cahier des charges.

3.4. Guide de déploiement et d'utilisation

3.4.1. Guide de déploiement

La base de données est **stable**. Elle a été conçue pour supporter le workflow complet : du dépôt de mémoire par l'étudiant à la planification par l'administrateur, en respectant les tables essentielles (Utilisateur, Étudiant, Enseignant, Soumissions, Soutenance) définies dans le cahier des charges.

3.4.2. Guide d'utilisation

Le guide est centralisé dans le fichier SUBMIT_TRAVAIL_GUIDE.md. Il détaille :

- La procédure de **soumission des documents** (mémoires PDF de moins de 50 Mo).
- Le suivi des **statuts de validation** (Soumis, Accepté, Rejeté) par l'étudiant.
- L'accès au **Dashboard de supervision** pour l'administrateur et l'enseignant.

4-Rapport de rencontre avec le professeur

Date	Nom du Professeur rencontré	Activités
15-12-2025	Mr AKAKPO	Présentation du cahier de charges, envoi vers Monsieur CESSI pour avoir plus de détails sur les besoins réels de la plateforme
16-12-2025	Mr CESSI	Explications en détails sur le projet, recueil des besoins du projet et plus d'explications sur comment concevoir la plateforme

5 - Conclusion

La réalisation du projet **Thesis Manager** progresse conformément au planning initial établi pour ce projet tuteuré. L'implémentation de l'architecture technique, combinant la puissance de **Next.js** pour le frontend et la précision de **GraphQL** pour la gestion des données, a permis de poser les bases d'une plateforme robuste, sécurisée et parfaitement adaptée aux exigences de l'Université de Lomé.

L'adoption de technologies de pointe telles que **Relay** pour la synchronisation des données et **Tamagui** pour l'interface utilisateur garantit une application non seulement performante, mais aussi évolutive et maintenable à long terme. La structuration rigoureuse des schémas GraphQL et la mise en place de scripts d'automatisation (setup-submit-travail.sh) assurent une traçabilité et une fiabilité répondant directement à la problématique de digitalisation des processus de soutenance.

Les prochaines étapes se concentreront sur la finalisation du module de **gestion documentaire** pour le dépôt sécurisé des mémoires (format PDF, max 50 Mo) ainsi que sur l'intégration complète des tableaux de bord destinés aux enseignants et aux administrateurs. L'objectif final demeure de fournir un outil clé en main facilitant la coordination entre tous les acteurs du parcours académique de l'Ecole Polytechnique de Lomé.

<<Merci>>