

جامعة شعيب الدكالي



universite chouaib doukkali

Application Web pour la gestion du service recherche

Réalisé par :
BADIAA ESSALIHI

Supervisé par :
Mr.boutkhoum

Juillet 2024

Table des matières

1 Conception	1
1.1 Analyse des besoins	1
1.1.1 Introduction	1
1.1.1.1 Contexte	1
1.1.1.2 Objectifs	1
1.1.2 Description des Espaces Utilisateurs	1
1.1.2.1 Espace Doctorant	1
1.1.2.2 Espace Enseignant	1
1.1.2.3 Espace Administrateur	2
1.2 Conception de la Base de Données	2
1.2.1 Table	2
1.2.1.1 Table : Utilisateurs	2
1.2.1.2 Table : Doctorants	3
1.2.1.3 Table : Infos Personnelles	3
1.2.1.4 Table : Infos Baccalauréat	3
1.2.1.5 Table : Infos Universitaires	3
1.2.1.6 Table : Infos Cycle Doctoral	3
1.2.1.7 Table : Enseignants	4
1.2.1.8 Table : Administrateurs	4
1.2.1.9 Table : Projets et Thèses	4
1.2.1.10 Table : Demandes	4
1.2.1.11 Table : Événements	4
1.2.1.12 Table : Notifications	4
1.2.2 Relations Clés	4
1.2.3 Modèle proposé	5
1.3 Les maquettes	6
2 Logiciels et Outils	9
2.1 Figma	9
2.2 Visual Studio Code	9
2.3 GitHub	9
2.4 StarUML	10
2.5 XAMPP	10
2.6 Postman	10
3 Technologies Utilisées	11
3.1 Front-end	11
3.2 Back-end	14
4 Réalisation	15
4.1 Architecture de l'Application	15
4.2 Frontend	16
4.2.1 Structure du Projet	16
4.2.2 Présentation des interfaces	17
4.2.2.1 Espace Doctorant	17
4.2.2.2 Espace Enseignant	21
4.2.2.3 Espace Administrateur	23
4.3 Backend	24
4.3.1 Structure du Projet	24
4.3.2 Backend test avec Postman	25
5 Conclusion	28
6 Références	29

Table des figures

1	Modèle proposé	5
2	Page d'accueil	6
3	Choix de profil	7
4	Page de connexion	7
5	Page d'inscription	8
6	Page de réinitialisation de mot de passe	8
7	Le logo de Figma	9
8	Le logo de Visual Studio Code	9
9	Le logo de Figma	10
10	Le logo de StarUML	10
11	Le logo de XAMPP	10
12	Le logo de Postman	11
13	Logo de React.js	11
14	Logo de Redux	11
15	Logo de Material-UI	11
16	Logo de Chakra UI	12
17	Logo de Bootstrap	12
18	Logo de Tailwind CSS	12
19	Logo d'Axios	12
20	Logo de React Router	13
21	Logo de Formik	13
22	Logo de Chart.js	13
23	EXemple de React Big Calendar	14
24	Logo de Node.js	14
25	Logo d'Express.js	14
26	Logo de Sequelize	14
27	Logo de Nodemailer	15
28	Le logo de phpMyAdmin	15
29	Diagramme d'architecture de l'application.	16
30	Structure du projet frontend.	16
31	Interface de l'espace Doctorant	17
32	Interface de l'espace Doctorant	18
33	Interface de partage de document	19
34	Interface de gestion des informations du doctorant	19
35	Interface de gestion des demandes du doctorant	20
36	Interface du Calendrier des Réunions	21
37	Interface du Calendrier des Événements	21
38	Interface de gestion des réunions pour les enseignants	22
39	Liste des doctorants et leurs sujets de recherche	22
40	Liste des doctorants et leurs sujets de recherche	23
41	Répartition des Doctorants par Spécialisation	23
42	Tableau des demandes des doctorants	24
43	Structure du projet backend.	24
44	Test d'affichage de tous les utilisateurs.	25
45	Test de la fonctionnalité de login.	26
46	Test de la fonctionnalité d'inscription.	26
47	Test de réinitialisation du mot de passe.	27
48	Test d'insertion des informations.	27

Introduction

Dans un monde de plus en plus connecté, le développement d'applications web joue un rôle crucial dans la transformation digitale des entreprises et des institutions. Ce projet vise à concevoir et à réaliser une application web moderne destinée aux étudiants en doctorat, aux enseignants et aux administrateurs. L'objectif principal est de fournir une plateforme intégrée qui facilite la gestion des informations académiques, la communication entre les utilisateurs, ainsi que l'organisation des projets et des événements.

La plateforme est conçue pour répondre aux besoins spécifiques des utilisateurs, en leur permettant de gérer efficacement leurs documents, d'accéder à des ressources académiques et de suivre leurs progrès au sein de leur parcours doctoral. En intégrant des fonctionnalités telles que l'authentification, la gestion des notifications, et un tableau de bord personnalisé, l'application offre une expérience utilisateur fluide et intuitive.

Pour atteindre ces objectifs, nous avons choisi d'utiliser des technologies modernes et performantes. Le frontend de l'application est construit avec **React.js**, une bibliothèque JavaScript réputée pour sa capacité à créer des interfaces utilisateur dynamiques et réactives. Pour la gestion de l'état de l'application, nous avons intégré **Redux**, facilitant la synchronisation des données entre les composants. Le style visuel est enrichi par des frameworks tels que **Material-UI** et **Bootstrap**, garantissant une interface esthétique et responsive.

Côté backend, nous avons opté pour **Node.js** et **Express.js**, offrant une architecture performante pour la création d'API RESTful. **Sequelize** est utilisé pour interagir avec notre base de données relationnelle, simplifiant la gestion des données et des modèles. Enfin, des outils comme **Nodemailer** et **phpMyAdmin** sont intégrés pour améliorer la gestion des e-mails et l'administration de la base de données.

Cette application ne se contente pas de répondre aux besoins actuels des utilisateurs, elle est également conçue pour être évolutive, permettant l'ajout de nouvelles fonctionnalités à l'avenir. Ce rapport présente le processus de développement de l'application, les technologies utilisées, ainsi que les défis rencontrés et les solutions mises en place.

1 Conception

1.1 Analyse des besoins

1.1.1 Introduction

1.1.1.1 Contexte

L'administration souhaite développer une application web pour optimiser la gestion et la communication entre les doctorants, les enseignants et les administrateurs. L'application proposée offrira trois espaces distincts, chacun étant conçu pour répondre aux besoins spécifiques des différents utilisateurs de la plateforme.

1.1.1.2 Objectifs

- **Facilitation de la communication :** Améliorer le partage d'informations entre doctorants et enseignants pour favoriser la collaboration et le suivi académique.
- **Gestion efficace des activités académiques :** Offrir des outils pour la gestion des projets de recherche, des thèses et des publications, permettant ainsi un suivi rigoureux et structuré.
- **Gestion efficace des activités académiques :** Offrir des outils pour la gestion des projets de recherche, des thèses et des publications, permettant ainsi un suivi rigoureux et structuré.

1.1.2 Description des Espaces Utilisateurs

1.1.2.1 Espace Doctorant

Fonctionnalités

- **Inscription et Authentification :**
 - **Inscription avec validation par email :** Les doctorants peuvent s'inscrire en utilisant leur adresse email, suivie d'une validation pour assurer la véracité des informations fournies.
 - **Authentification via email et mot de passe :** Accès sécurisé à l'application.
 - **Récupération de mot de passe :** Option pour récupérer ou réinitialiser le mot de passe en cas de perte.
- **Tableau de Bord :**
 - **Édition de documents :** Les doctorants peuvent éditer divers documents administratifs, tels que des demandes de changement de sujet de thèse, des retraits provisoires ou définitifs de diplômes, des demandes d'attestation d'inscription, des conventions de stage, et des cartes d'étudiant.
 - **Gestion des informations personnelles :** Interface permettant de gérer les informations personnelles, les données du baccalauréat, le cursus universitaire, et les inscriptions au cycle doctoral.
 - **Calendrier :** Affichage des formations et des événements académiques.
 - **Pré-soutenance et post-soutenance :** Édition du dossier administratif de pré-soutenance et des documents nécessaires après la soutenance.
 - **Demandes spécifiques :** Soumission de demandes d'inscription et de dérogations, et upload de demandes signées.

Interface Utilisateur

- **Interface intuitive et conviviale :** Design ergonomique facilitant l'utilisation par tous les doctorants.
- **Design responsive :** Accessibilité optimisée pour une utilisation sur mobile et tablette, assurant une flexibilité d'accès en tout lieu.

1.1.2.2 Espace Enseignant

Fonctionnalités

- **Inscription et Authentification :**
 - **Inscription avec validation par email :** Inscription basée sur une liste pré-établie pour garantir que seuls les enseignants autorisés peuvent accéder à l'application.
 - **Authentification via email et mot de passe :** Sécurisation de l'accès.

- **Tableau de Bord :**
 - **Vue d'ensemble des doctorants encadrés** : Suivi des doctorants sous la supervision de l'enseignant.
 - **Calendrier des soutenances et réunions** : Planification et gestion des soutenances et des réunions académiques.
 - **Diffusion d'informations** : Partage des informations sur les événements, les formations, et les appels à projet.
- **Gestion des Projets et Thèses :**
 - **Suivi de l'avancement** : Outil de suivi de l'état d'avancement des projets de recherche et des thèses.
 - **Commentaires et feedbacks** : Possibilité de commenter et de fournir des retours sur les documents soumis par les doctorants.

Interface Utilisateur

- **Interface intuitive et conviviale** : Simplifie l'interaction avec la plateforme.
- **Design responsive** : Adaptabilité à différents dispositifs pour une utilisation flexible.

1.1.2.3 Espace Administrateur

Fonctionnalités

- **Inscription et Authentification :**
 - **Authentification via email et mot de passe** : Assure la sécurité des accès administratifs.
- **Tableau de Bord :**
 - **Vue d'ensemble des activités** : Aperçu complet des activités des doctorants et des enseignants.
 - **Notifications et alertes** : Gestion des notifications importantes et des alertes pour un suivi proactif.
 - **Paramétrage de l'application** : Introduction des données nécessaires pour configurer et personnaliser l'application selon les besoins institutionnels.
 - **Validation des demandes** : Processus de validation pour les demandes soumises par les enseignants et les doctorants.
- **Gestion des Utilisateurs :**
 - **Création, modification et suppression de comptes** : Gestion complète des comptes utilisateurs.
 - **Gestion des droits et permissions** : Attribution et gestion des rôles et permissions pour assurer un accès contrôlé et sécurisé.
- **Supervision des Projets et Thèses :**
 - **Suivi de l'avancement** : Surveillance de l'évolution des projets et des thèses.
 - **Gestion des soutenances et des réunions** : Organisation et gestion des sessions de soutenance et des réunions académiques.
- **Communication :**
 - **Envoi de notifications et d'annonces** : Facilitation de la communication institutionnelle par des notifications et des annonces.

Interface Utilisateur

- **Interface intuitive et conviviale** : Conçue pour une utilisation facile et efficace par les administrateurs.
- **Design responsive** : Accessibilité sur divers dispositifs pour une gestion flexible.
- **Tableau de bord personnalisé** : Personnalisation des informations affichées selon les besoins de l'utilisateur administratif.

1.2 Conception de la Base de Données

1.2.1 Table

1.2.1.1 Table : Utilisateurs

La table Utilisateurs contient les informations de base sur tous les utilisateurs de l'application.

- **id_utilisateur (PK)** : Identifiant unique pour chaque utilisateur.
- **email (Unique)** : Adresse email de l'utilisateur, utilisée pour la connexion.
- **mot_de_passe_hache** : Mot de passe hashé pour des raisons de sécurité.
- **role** : Rôle de l'utilisateur dans l'application (Doctorant, Enseignant, Administrateur).
- **prenom** : Prénom de l'utilisateur.
- **nom** : nom de l'utilisateur.
- **tele** : Numéro de téléphone de l'utilisateur.

1.2.1.2 Table : Doctorants

- **id_doctorant (PK)** : Identifiant unique pour chaque doctorant.
- **id_utilisateur (FK - Utilisateurs)** : Référence à l'utilisateur associé.

1.2.1.3 Table : Infos Personnelles

La table infos_personnelles contient des informations personnelles détaillées des doctorants.

- **id_doctorant (FK - Doctorants)** : Référence au doctorant associé.
- **civilite** : Civilité (Monsieur, Madame, etc.).
- **sexe** : Sexe du doctorant.
- **nom** : Nom du doctorant.
- **prenom** : Prénom du doctorant.
- **date_naissance** : Date de naissance du doctorant.
- **email** : Adresse email du doctorant.
- **telephone** : Numéro de téléphone du doctorant.
- **lieu_naissance** : Lieu de naissance du doctorant.
- **cnie** : Numéro de la carte nationale d'identité.
- **situation_socioprofessionnelle** : Situation socioprofessionnelle actuelle.

1.2.1.4 Table : Infos Baccalauréat

La table infos_baccalaureat contient des informations sur le baccalauréat des doctorants.

- **id_doctorant (FK - Doctorants)** : Référence au doctorant associé.
- **annee_bac** : Année d'obtention du baccalauréat.
- **type_bac** : Type de baccalauréat (Scientifique, Littéraire, etc.).
- **mention** : Mention obtenue au baccalauréat.
- **cne_massar** : Numéro national de l'étudiant (CNE) ou code Massar.
- **serie_bac** : Série du baccalauréat.
- **academie** : Académie où le baccalauréat a été obtenu.
- **province** : Province où le baccalauréat a été obtenu.

1.2.1.5 Table : Infos Universitaires

La table infos_universitaires contient des informations sur les études universitaires des doctorants.

- **id_doctorant (FK - Doctorants)** : Référence au doctorant associé.
- **diplome** : Diplôme obtenu.
- **etablissement** : Établissement où le diplôme a été obtenu.
- **universite** : Université où le diplôme a été obtenu.

1.2.1.6 Table : Infos Cycle Doctoral

La table infos_cycle_doctoral contient des informations sur le cycle doctoral des doctorants.

- **id_doctorant (FK - Doctorants)** : Référence au doctorant associé.
- **formation_doctorale** : Formation doctorale suivie.
- **annee_soutenance_prevue** : Année prévue pour la soutenance de la thèse.
- **specialite** : Spécialité de recherche.
- **directeur_these** : Directeur de thèse.
- **structure_recherche_directeur** : Structure de recherche du directeur de thèse.
- **co_directeur_these** : Co-directeur de thèse (le cas échéant).

- **structure_recherche_co_directeur** : Structure de recherche du co-directeur de thèse.
- **universite_cotutelle** : Université de cotutelle (le cas échéant).
- **sujet_recherche** : Sujet de recherche de la thèse.

1.2.1.7 Table : Enseignants

La table Enseignants contient des informations spécifiques aux enseignants.

- **id_enseignant (PK)** : Identifiant unique pour chaque enseignant.
- **id_utilisateur (FK - Utilisateurs)** : Référence à l'utilisateur associé.
- **liste_pre_etablie** : Liste des doctorants encadrés.

1.2.1.8 Table : Administrateurs

- **id_administrateur (PK)** : Identifiant unique pour chaque administrateur.
- **id_utilisateur (FK - Utilisateurs)** : Référence à l'utilisateur associé.

1.2.1.9 Table : Projets et Thèses

- **id_projet (PK)** : Identifiant unique pour chaque projet.
- **titre** : Titre du projet.
- **description** : Description détaillée du projet.
- **statut** : Statut du projet (en cours, terminé, etc.).
- **date_debut** : Date de début du projet.
- **date_fin** : Date de fin prévue du projet.
- **id_doctorant (FK - Doctorants)** : Référence au doctorant associé.
- **id_enseignant (FK - Enseignants)** : Référence à l'enseignant encadrant.

1.2.1.10 Table : Demandes

- **id_demande (PK)** : Identifiant unique pour chaque demande.
- **type_demande** : Type de demande (administrative, recherche, autre).
- **statut** : Statut de la demande (soumise, en cours, validée, rejetée).
- **date_soumission** : Date de soumission de la demande.
- **id_doctorant (FK - Doctorants)** : Référence au doctorant ayant soumis la demande.
- **id_enseignant (FK - Enseignants, nullable)** : Référence à l'enseignant validant (le cas échéant).
- **détails** : Détails supplémentaires sur la demande.

1.2.1.11 Table : Événements

- **id_evenement (PK)** : Identifiant unique pour chaque événement.
- **titre** : Titre de l'événement.
- **description** : Description détaillée de l'événement.
- **date_evenement** : Date de l'événement.
- **lieu** : Lieu de l'événement.
- **type** : Type d'événement (conférence, atelier, etc.).

1.2.1.12 Table : Notifications

- **id_notification (PK)** : Identifiant unique pour chaque notification.
- **message** : Contenu du message de la notification.
- **date** : Date d'envoi de la notification.
- **id_utilisateur (FK - Utilisateurs)** : Référence à l'utilisateur recevant la notification.

1.2.2 Relations Clés

Les relations entre les différentes tables assurent la cohérence et l'intégrité des données :

- Chaque utilisateur (Utilisateurs) peut être associé à un doctorant (Doctorants), un enseignant (Enseignants) ou un administrateur (Administrateurs) via id_utilisateur.
- Les projets et thèses (Projets) sont liés à un doctorant (Doctorants) et un enseignant (Enseignants).

- Les demandes (Demandes) sont faites par les doctorants et validées par les enseignants ou les administrateurs.
- Les événements (Événements) sont organisés par les utilisateurs et peuvent être ajoutés au calendrier personnel des utilisateurs.
- Les notifications (Notifications) sont envoyées aux utilisateurs pour les informer des mises à jour importantes.

1.2.3 Modèle proposé

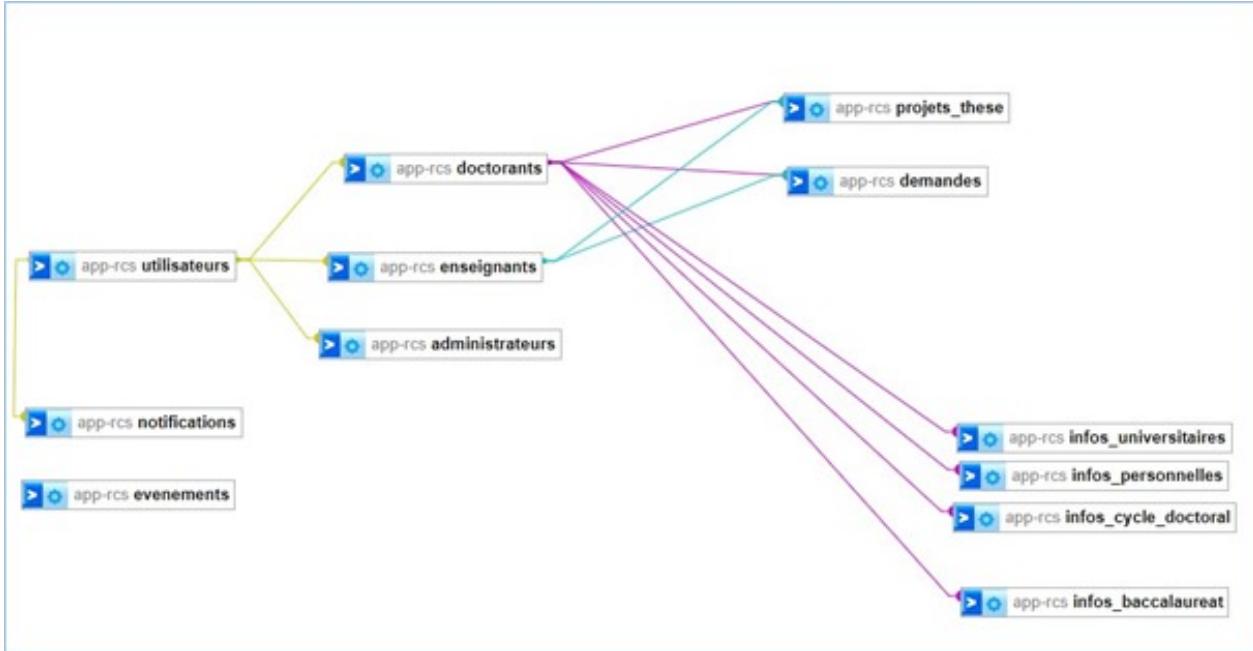


FIGURE 1 – Modèle proposé

1.3 Les maquettes



كلية العلوم
+٢٤٣٦١١٢٠١٤١
FACULTÉ DES SCIENCES

Meilleure plateforme
pour un rendement
optimal.

[Se connecter](#)

[Pas encore inscrit ?](#)



Centre des Etudes Doctorales
CED

 olimaclientservice@gmail.com  olimaclientservice@gmail.com © olima tous droits réservés 2024

FIGURE 2 – Page d'accueil



Votre profile ?



Doctorant



Enseignant



Administrateur

FIGURE 3 – Choix de profil

FIGURE 4 – Page de connexion



جامعة شعيب الدكالي

Inscription

PRENOM
Badiaa

NOM
ESSALIHI

EMAIL
nomprenom@example.com

MOT DE PASSE

S'inscrire

FIGURE 5 – Page d’inscription



جامعة شعيب الدكالي

Réinitialiser votre mot de passe

VOTRE EMAIL ?
nomprenom@example.com

Récupérer le !

FIGURE 6 – Page de réinitialisation de mot de passe

2 Logiciels et Outils

2.1 Figma

Figma est une plateforme de conception d’interfaces utilisateur (UI) et d’expérience utilisateur (UX) basée sur le cloud. C’est un outil largement adopté par les concepteurs, développeurs et équipes de design pour créer, collaborer et prototyper des interfaces utilisateur interactives pour des applications, sites web, et autres produits numériques. Grâce à son interface intuitive et conviviale, Figma permet aux utilisateurs de créer facilement des maquettes, wireframes, et prototypes interactifs. L’un des grands avantages de Figma est qu’il est entièrement basé sur le cloud, ce qui facilite la collaboration en temps réel entre les équipes, peu importe leur localisation. Cette fonctionnalité est idéale pour la gestion de projets à distance et la coordination entre différents membres d’une équipe. De plus, la gestion des versions est simplifiée, permettant à chaque contributeur de suivre les modifications et d’apporter des ajustements sans complexité.

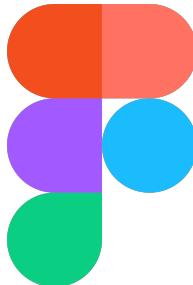


FIGURE 7 – Le logo de Figma

2.2 Visual Studio Code

Visual Studio Code (VSCode) est un éditeur de code source largement utilisé développé par Microsoft. Il est connu pour sa légèreté, ses options de personnalisation étendues et son support pour divers langages de programmation et technologies. VSCode propose des fonctionnalités telles que la mise en évidence du code, l’autocomplétion, les capacités de débogage, l’intégration de contrôle de version et un écosystème d’extensions qui permettent aux développeurs d’améliorer leur fonctionnalité en fonction de leurs besoins spécifiques.

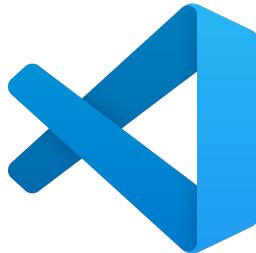


FIGURE 8 – Le logo de Visual Studio Code

2.3 GitHub

GitHub est une plateforme de développement collaboratif basée sur le système de contrôle de version Git. Elle permet aux développeurs de travailler ensemble sur des projets, de partager du code source, de suivre les modifications, de gérer les problèmes et les demandes de fusion, et de collaborer de manière transparente sur des projets logiciels. En utilisant GitHub, vous pouvez héberger vos dépôts de code source, collaborer avec d’autres développeurs en facilitant le travail en équipe, examiner les modifications apportées au code, signaler et résoudre des problèmes, et gérer le cycle de vie de développement d’un projet.

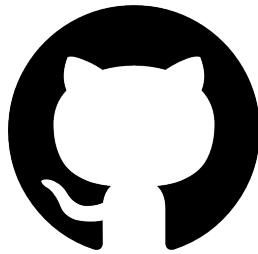


FIGURE 9 – Le logo de Figma

2.4 StarUML

StarUML est un logiciel de modélisation UML (Unified Modeling Language) largement utilisé pour concevoir et créer des diagrammes de modèles logiciels. Il permet aux développeurs et aux ingénieurs de créer des représentations visuelles des structures, des interactions et comportements des systèmes logiciels.



FIGURE 10 – Le logo de StarUML

2.5 XAMPP

XAMPP est une solution de serveur web open-source qui inclut Apache, MySQL (ou MariaDB), et PHP. Il permet de créer un environnement de développement local complet pour les projets web. XAMPP facilite la mise en place d'un serveur local pour le développement et le test des applications web avant leur déploiement sur un serveur de production.



FIGURE 11 – Le logo de XAMPP

2.6 Postman

Postman est un outil de développement API utilisé pour tester et déboguer les requêtes API. Il permet aux développeurs de créer et d'exécuter des requêtes HTTP, de vérifier les réponses des API, et de gérer les collections de requêtes pour le développement et le test des APIs. Postman est particulièrement utile pour tester les endpoints API et s'assurer que les services fonctionnent comme prévu avant l'intégration avec l'application frontend.



FIGURE 12 – Le logo de Postman

3 Technologies Utilisées

3.1 Front-end

- **React.js** : Une bibliothèque JavaScript développée par Facebook pour construire des interfaces utilisateur dynamiques et interactives. Elle permet de créer des composants réutilisables et de gérer l'état de l'application efficacement. React.js est souvent utilisée pour le développement d'interfaces utilisateur modernes et performantes.



FIGURE 13 – Logo de React.js

- **Redux** : Une bibliothèque JavaScript utilisée pour la gestion d'état global dans les applications. Redux centralise l'état dans un store unique, facilitant ainsi la gestion et la synchronisation des données entre les composants.



FIGURE 14 – Logo de Redux

- **Material-UI** : Framework de composants React qui implémente les principes de Material Design. Il fournit une collection de composants stylisés pour la création d'interfaces élégantes et cohérentes.

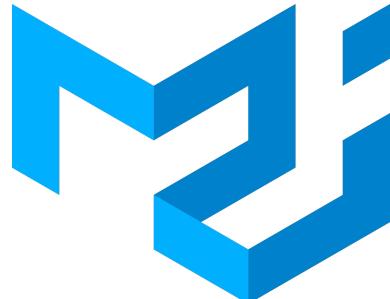


FIGURE 15 – Logo de Material-UI

- **Chakra UI** : Une bibliothèque de composants React accessible et personnalisable, axée sur la simplicité et la rapidité pour créer des interfaces utilisateur modernes.

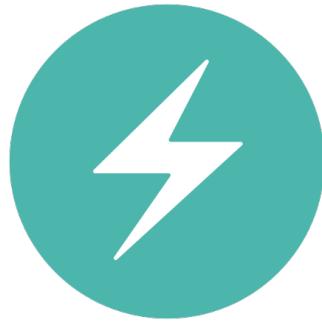


FIGURE 16 – Logo de Chakra UI

- **Bootstrap** : Framework CSS populaire pour créer des sites web réactifs et des interfaces élégantes grâce à des composants et des styles préconçus.

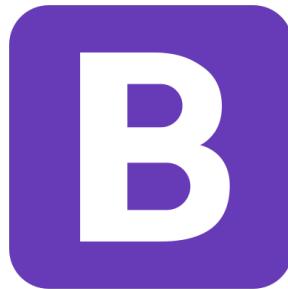


FIGURE 17 – Logo de Bootstrap

- **Tailwind CSS** : Un framework CSS utilitaire permettant une grande flexibilité et une personnalisation rapide des styles, facilitant une approche modulaire et maintenable pour le design web.

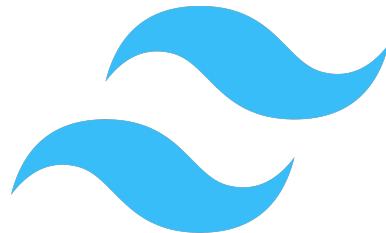


FIGURE 18 – Logo de Tailwind CSS

- **Axios** : Client HTTP basé sur les promesses, utilisé pour effectuer des requêtes API entre le frontend et le backend, permettant des communications réseau efficaces.



FIGURE 19 – Logo d'Axios

- **React Router** : Librairie de gestion de navigation pour React, permettant de gérer les différentes vues de l'application en fonction des URL de manière dynamique.

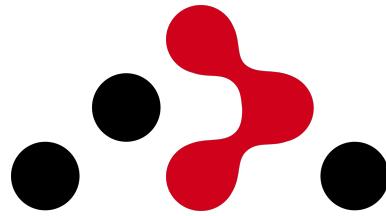


FIGURE 20 – Logo de React Router

- **Formik** : Librairie pour la gestion des formulaires dans React, simplifiant la gestion des états, des validations et des soumissions.



FIGURE 21 – Logo de Formik

- **Chart.js** : Librairie pour la création de graphiques interactifs, permettant d'afficher des données sous forme de graphiques à barres, lignes, secteurs, etc.



FIGURE 22 – Logo de Chart.js

- **React Big Calendar** : Composant de calendrier pour React, permettant de gérer des événements avec des vues mensuelles, hebdomadaires et journalières.

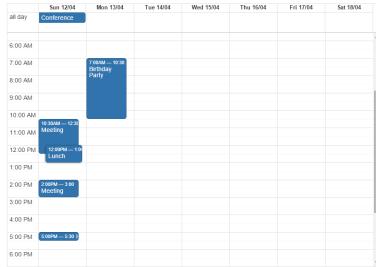


FIGURE 23 – EXemple de React Big Calendar

3.2 Back-end

- **Node.js** : Environnement d'exécution JavaScript côté serveur, permettant de construire des applications backend performantes, notamment pour gérer des I/O intensives et des opérations asynchrones.



FIGURE 24 – Logo de Node.js

- **Express.js** : Framework minimaliste pour Node.js, utilisé pour créer des APIs RESTful et gérer les routes dans les applications backend.



FIGURE 25 – Logo d'Express.js

- **Sequelize** : ORM pour Node.js qui permet d'interagir avec des bases de données relationnelles de manière abstraite, facilitant la gestion des modèles et des relations entre les données.



FIGURE 26 – Logo de Sequelize

- **Nodemailer** : Librairie pour l'envoi d'e-mails depuis des applications Node.js via SMTP, souvent utilisée pour l'envoi de notifications et de communications automatisées.



FIGURE 27 – Logo de Nodemailer

- **phpMyAdmin** : phpMyAdmin est un outil libre et open-source écrit en PHP pour gérer MySQL et MariaDB via une interface web. Il permet aux utilisateurs d'administrer facilement les bases de données, d'exécuter des requêtes SQL, de gérer les utilisateurs, les privilèges, les structures de tables, et d'importer ou exporter des données.



FIGURE 28 – Le logo de phpMyAdmin

4 Réalisation

4.1 Architecture de l’Application

L’application que j’ai développée repose sur une architecture web moderne combinant **React** pour le frontend et **Node.js/Express** pour le backend. La communication entre ces deux parties est orchestrée par des requêtes HTTP via une API REST, et la gestion des sessions est sécurisée grâce à des **tokens JWT** (JSON Web Tokens).

Le frontend en **React** utilise des composants dynamiques pour gérer l’interface utilisateur, avec **Axios** comme bibliothèque pour envoyer des requêtes HTTP. Les actions de l’utilisateur, comme la soumission d’un formulaire de connexion ou la gestion des projets, sont transmises au backend via des services. Le **router React** assure la navigation fluide entre les différentes sections de l’application.

Du côté du backend, j’ai utilisé **Node.js** et le framework **Express** pour gérer les requêtes entrantes. Le routeur Express redirige les requêtes vers les contrôleurs appropriés, qui contiennent la logique métier. Ces contrôleurs utilisent **Sequelize**, un ORM (Object-Relational Mapping), pour interagir avec la base de données **MySQL**. Sequelize facilite la gestion des données en permettant de manipuler les entités sous forme de modèles. Par exemple, les entités telles que **Utilisateurs** et **Projets** sont représentées par des modèles Sequelize, ce qui permet de réaliser des opérations CRUD (Création, Lecture, Mise à jour, Suppression) en toute simplicité.

La base de données **MySQL** stocke toutes les informations relatives aux utilisateurs, projets, et autres entités de l’application. Sequelize permet de définir des relations entre ces entités et d’effectuer des requêtes complexes tout en gardant une abstraction de la base de données sous-jacente.

Enfin, la sécurité de l'application est assurée par l'utilisation de **tokens JWT**. Lorsqu'un utilisateur se connecte, un token JWT est généré et stocké dans le frontend pour authentifier les requêtes ultérieures. Le backend vérifie la validité de ce token avant d'autoriser l'accès aux ressources protégées, comme la liste des projets.

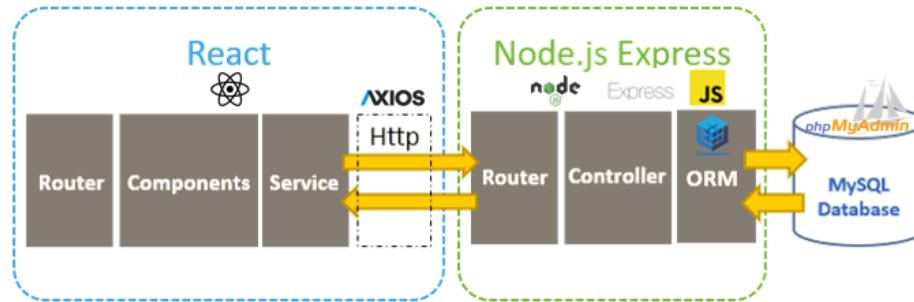


FIGURE 29 – Diagramme d'architecture de l'application.

Cette architecture permet de garantir une interaction fluide et sécurisée entre les différents composants de l'application, tout en respectant les meilleures pratiques en matière de développement web.

4.2 Frontend

4.2.1 Structure du Projet

La structure du projet frontend en **React** est organisée de manière à faciliter le développement et la maintenance de l'application. Voici un aperçu des principaux dossiers et fichiers :

```

    < WebApp-RCS
      < frontend
        > node_modules
        > public
        < src
          < api
            JS axios.js
            JS userService.js
            > assets
            > components
            > utils
            # App.css
            JS App.js
            JS App.test.js
            # index.css
            JS index.js
            logo.svg
            JS reportWebVitals.js
            JS setupTests.js
            .env
            .gitignore
            {} package-lock.json
            {} package.json
            JS postcss.config.js
            README.md

```

FIGURE 30 – Structure du projet frontend.

- **node modules/** : Contient toutes les dépendances et packages installés.

- **public/** : Contient des fichiers statiques comme `index.html` et d'autres ressources publiques.
- **src/** : Le dossier source contenant la logique principale de l'application.
 - **api/** :
 - `axios.js` : Gère les requêtes API avec Axios.
 - `userService.js` : Services API pour les utilisateurs (connexion, inscription, etc.).
 - **assets/** : Dossier contenant les images, polices, et autres ressources statiques.
 - **components/** : Composants React réutilisables dans l'application.
 - **utils/** : Fonctions utilitaires réutilisables dans plusieurs parties de l'application.
- `App.css` : Styles pour le composant App.
- `App.js` : Composant principal englobant toute l'application.
- `index.js` : Point d'entrée de l'application React pour le rendu dans le DOM.
- `README.md` : Documentation du projet expliquant l'installation et l'utilisation.
Voici d'autres fichiers importants :
- `.env` : Variables d'environnement (comme les clés API).
- `.gitignore` : Liste des fichiers à ignorer par Git (`node_modules`, `.env`).
- `package.json` : Liste des dépendances et scripts pour les tâches courantes.
Cette organisation garde le code propre et modulaire, facilitant la maintenance et l'extensibilité.

4.2.2 Présentation des interfaces

L'application comporte trois espaces principaux, chacun conçu pour répondre aux besoins spécifiques de ses utilisateurs : le doctorant, l'enseignant, et l'administrateur. Chaque espace offre des fonctionnalités et des interfaces adaptées aux rôles et responsabilités de ses utilisateurs respectifs.

4.2.2.1 Espace Doctorant

Le tableau de bord de l'espace Doctorant offre un accès centralisé à plusieurs fonctionnalités essentielles pour les doctorants. Il inclut des sections telles que les Documents Partagés, où les doctorants peuvent consulter les fichiers mis à disposition par leurs encadrants ou administrateurs, ainsi qu'une section Documents, leur permettant de gérer leurs propres documents. La section Demandes permet aux doctorants de soumettre des requêtes administratives ou académiques. En outre, le tableau de bord comprend un Calendrier, pour suivre les échéances importantes, ainsi que la section Événements, où sont affichés les événements pertinents tels que des conférences ou des séminaires.

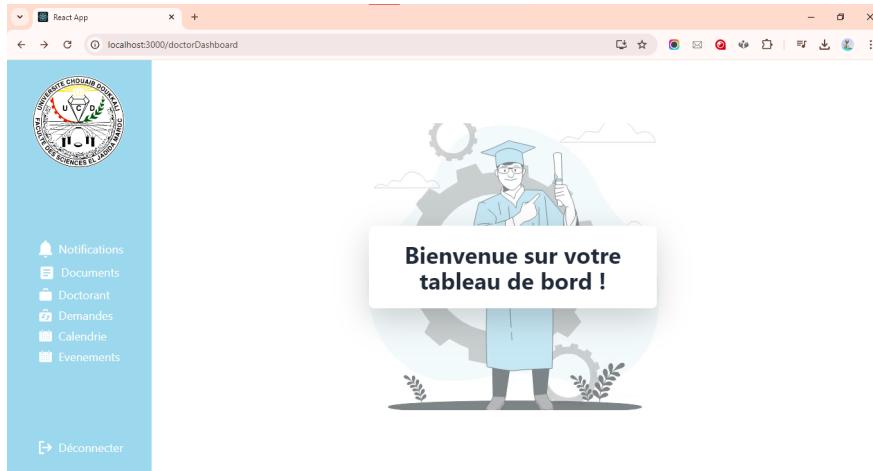


FIGURE 31 – Interface de l'espace Doctorant

Documents Partagés : Cette section présente une liste des documents partagés avec le doctorant, soit par les enseignants, soit par l'administration.

- **Liste des Documents Partagés** : Affichage d'une liste des documents partagés avec le doctorant, incluant des informations telles que le nom du document, la personne qui l'a partagé, et la date de partage.
- **Fonctionnalités** : Chaque document indique son nom, la personne qui l'a partagé, et la date de partage. Un lien ou un bouton permet le téléchargement direct du document.
- **Options de Recherche et Visualisation** : Il est possible de rechercher un document par le nom de l'émetteur ou par la date de partage. De plus, une frise chronologique permet de visualiser les documents partagés par semaine.

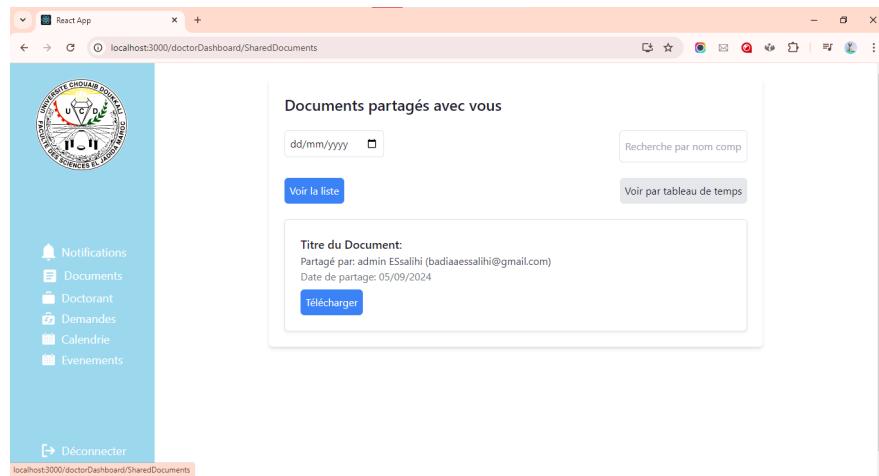


FIGURE 32 – Interface de l'espace Doctorant

Partage de Document : Cette section permet aux utilisateurs de partager des documents avec d'autres acteurs de la plateforme, tels que les doctorants, enseignants ou administrateurs.

- **L'utilisateur remplit le formulaire de partage de document :**
 - L'utilisateur entre le titre du document et sélectionne un fichier à télécharger.
 - L'utilisateur choisit le rôle du récepteur du document (doctorant, administrateur, enseignant).
 - Si l'utilisateur choisit doctorant ou enseignant, il doit d'abord sélectionner un département pour filtrer la liste des récepteurs disponibles.
 - Si l'utilisateur choisit administrateur, il n'est pas nécessaire de sélectionner un département.
 - L'utilisateur sélectionne ensuite un récepteur spécifique à partir de la liste filtrée (doctorant ou enseignant pour les départements, ou un administrateur).
- **Upload du fichier :** Une fois le formulaire complété, l'utilisateur soumet le fichier à l'application.
- **Création et partage du document :** Le document est associé aux informations fournies (titre, fichier, récepteur), et un retour de confirmation est envoyé à l'utilisateur si le partage est réussi.

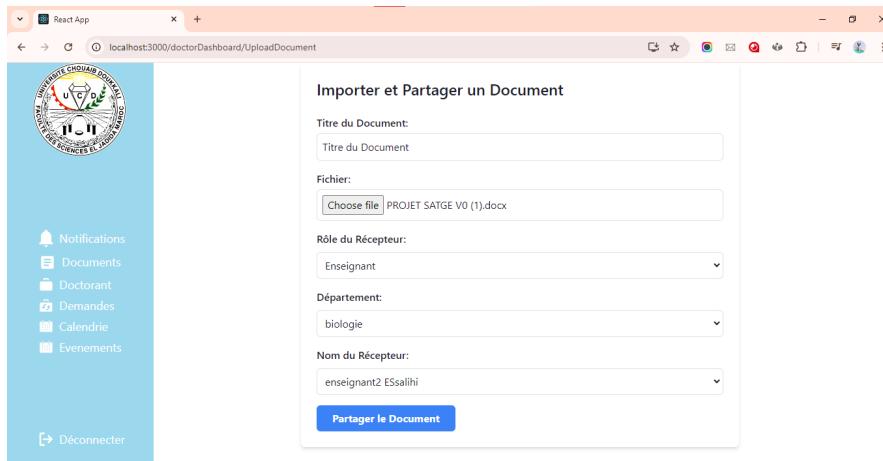


FIGURE 33 – Interface de partage de document

Doctorant : Cette section permet au doctorant de gérer ses informations personnelles et académiques. Initialement, les formulaires sont vides, et c'est au doctorant de les compléter avec ses données personnelles. Chaque sous-section correspond à un type spécifique d'information, et les données peuvent être modifiées à tout moment.

- **Informations Personnelles :** Le doctorant remplit ses informations de base, telles que son nom, prénom, date de naissance, et adresse. Ces informations sont essentielles pour l'identification dans le système.
- **Informations sur le Baccalauréat :** Cette sous-section permet au doctorant de saisir les détails de son baccalauréat, incluant l'année d'obtention, la spécialité, et l'établissement d'obtention.
- **Informations Universitaires :** Le doctorant fournit ici les informations relatives à son parcours universitaire, telles que les diplômes obtenus, les universités fréquentées, et les mentions obtenues durant ses études.
- **Informations sur le Cycle Doctoral :** Cette section est dédiée aux informations spécifiques au cycle doctoral du doctorant. Il peut y indiquer son sujet de thèse, son directeur de thèse, la date d'inscription en doctorat, ainsi que d'autres détails pertinents à son programme de recherche.
- **Modification des Informations :** À tout moment, le doctorant peut modifier ses informations via cette section. Des options d'édition sont disponibles pour chaque sous-section, permettant une mise à jour continue des données.

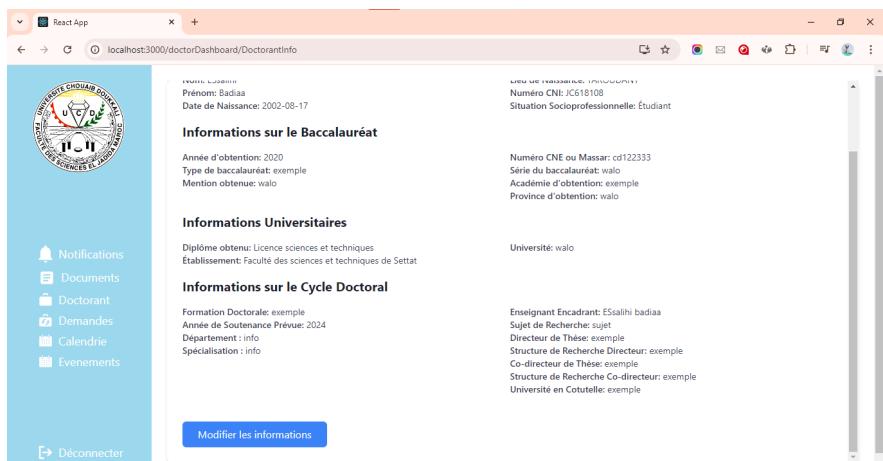


FIGURE 34 – Interface de gestion des informations du doctorant

Demandes : Cette section permet au doctorant de gérer ses demandes administratives et académiques. Les demandes soumises par le doctorant sont présentées sous forme de tableau, avec des informations

claires sur chaque requête, son statut et sa date de soumission. Le doctorant a également la possibilité d'ajouter de nouvelles demandes ou de modifier celles en cours.

- **Tableau des Demandes :** Le tableau affiche toutes les demandes effectuées par le doctorant, avec les colonnes suivantes :
 - *Type de Demande* : Il peut s'agir de demandes d'inscription, de dérogation, de tirage de thèse, etc.
 - *Statut de la Demande* : Le statut indique si la demande est en cours de traitement, acceptée ou rejetée.
 - *Date de Soumission* : La date à laquelle la demande a été soumise est affichée pour référence.
- **Ajout de Demandes :** Le doctorant peut soumettre de nouvelles demandes en remplissant un formulaire dédié. Il sélectionne le type de demande, entre une description si nécessaire, et soumet sa requête pour traitement par les responsables.
- **Modification des Demandes :** Tant que la demande n'a pas été traitée, le doctorant a la possibilité de la modifier. Cela inclut la mise à jour du type de demande, l'ajout d'informations supplémentaires, ou l'annulation de la requête.

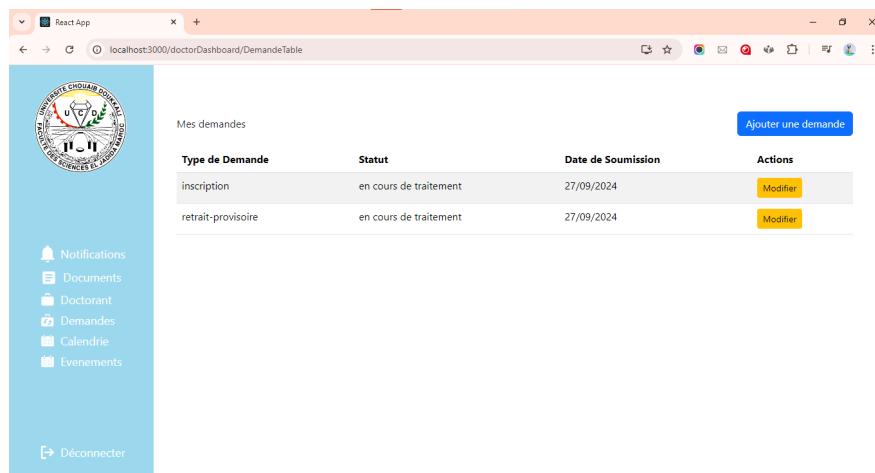


FIGURE 35 – Interface de gestion des demandes du doctorant

Calendrier des Réunions : Cette section permet aux doctorants de consulter et de gérer les réunions programmées via un calendrier interactif. Les réunions sont organisées selon différentes vues (mois, semaine, jour), et chaque réunion peut être consultée en détail.

- **Affichage du Calendrier :** Le calendrier offre plusieurs options de visualisation, permettant aux doctorants de consulter les réunions prévues selon leurs préférences :
 - *Vue Mois* : Permet une vue d'ensemble des réunions programmées sur un mois entier.
 - *Vue Semaine* : Affiche les réunions organisées sur une semaine donnée, facilitant la planification à court terme.
 - *Vue Jour* : Offre une vue détaillée des réunions prévues pour un jour spécifique.
- **Détails des Réunions :** En cliquant sur une réunion spécifique dans le calendrier, le doctorant peut accéder aux informations détaillées, telles que :
 - *Titre de la Réunion* : Sujet ou titre de la réunion.
 - *Date et Heure* : Date et heure exactes de la réunion.
 - *Participants* : Liste des participants attendus.
 - *Lieu ou Lien* : Lieu physique de la réunion ou lien vers une visioconférence en ligne.
 - *Description* : Informations supplémentaires sur l'objet ou les points à aborder lors de la réunion.

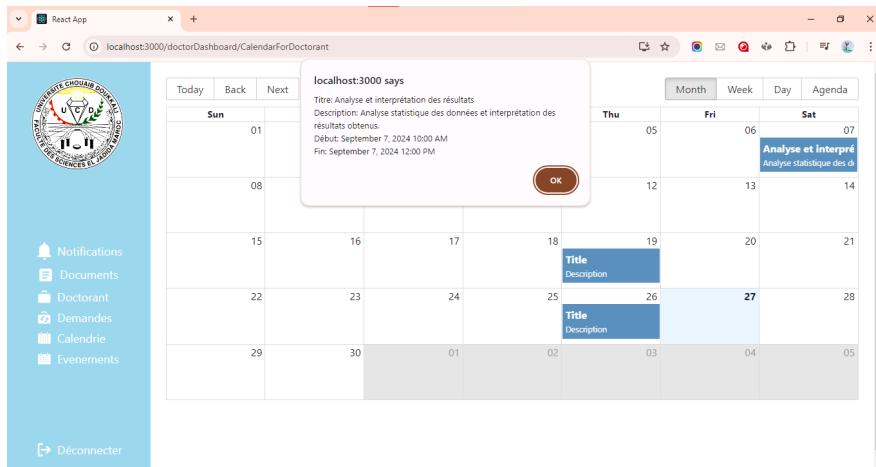


FIGURE 36 – Interface du Calendrier des Réunions

Calendrier des Événements Cette section permet aux doctorants de consulter et de gérer les événements à venir via un calendrier interactif. Les événements comprennent des formations, des séminaires, et des appels à projets

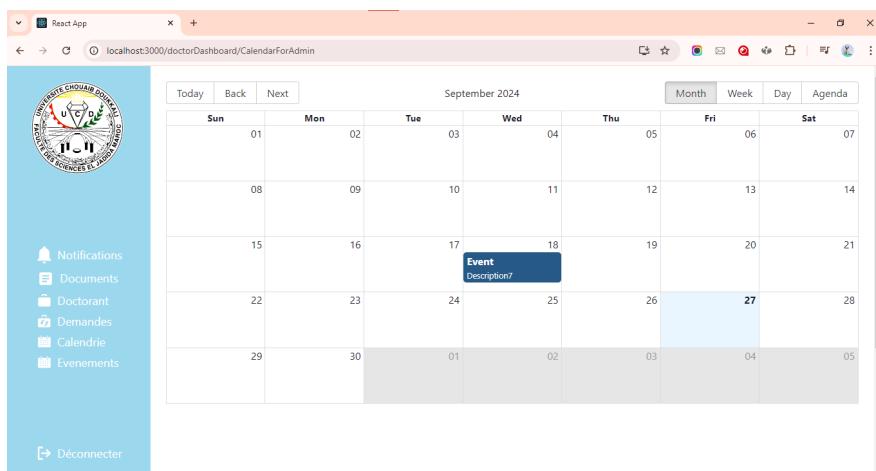


FIGURE 37 – Interface du Calendrier des Événements

4.2.2.2 Espace Enseignant

Le tableau de bord de l'espace Enseignant offre un accès centralisé à plusieurs fonctionnalités essentielles, similaires à celles de l'espace Doctorant. Il inclut des sections telles que les Documents Partagés, où les enseignants peuvent consulter les fichiers mis à disposition par leurs collègues ou administrateurs, ainsi qu'une section Documents pour gérer leurs propres documents. un Calendrier pour programmer des réunions avec leurs doctorants une section Événements, où sont affichés les événements pertinents tels que des conférences ou des séminaires. Une fonctionnalité supplémentaire, la Vue globale, offre aux enseignants une perspective d'ensemble sur les activités et les documents partagés au sein de leur département ou de l'institution.

Réunions Dans l'espace Enseignant, les enseignants ont la possibilité de programmer des réunions avec leurs doctorants. Cette fonctionnalité comprend un formulaire où l'enseignant peut spécifier plusieurs champs essentiels :

- **Titre** : Permet de définir un titre pour la réunion, facilitant l'identification du sujet.
- **Description** : Champ optionnel pour fournir des informations supplémentaires sur l'objet de la réunion.

- **Département** : L'enseignant doit d'abord sélectionner le département concerné pour filtrer la liste des doctorants disponibles.
 - **Doctorant** : Une fois le département sélectionné, l'enseignant choisit le doctorant concerné par la réunion.
 - **Date** : Sélection de la date à laquelle la réunion est prévue.
 - **Heure** : Indication de l'heure de début de la réunion.
- Les enseignants ont également la possibilité de modifier ou de supprimer une réunion déjà programmée, garantissant ainsi une flexibilité dans la gestion de leur emploi du temps.

La partie droite de l'image montre une interface de création de réunion. Elle comprend les champs suivants :

- Title**: champ de texte pour le titre de la réunion.
- Description**: champ de texte pour la description de la réunion.
- Date**: champ de texte avec la valeur "2024-09-27".
- Start Time**: champ de texte avec la valeur "00:00".
- End Time**: champ de texte avec la valeur "00:00".
- Select Department**: champ déroulant avec la valeur "Select a department".
- Select Doctorants**: champ déroulant avec la valeur "Select Doctorants".

À la base de l'interface, il y a deux boutons : "Create" et "Cancel".

FIGURE 38 – Interface de gestion des réunions pour les enseignants

Vue globale La section "Vue globale" permet aux enseignants d'avoir une vision d'ensemble des doctorants qu'ils encadrent. Cette fonctionnalité présente une liste exhaustive des doctorants associés à l'enseignant, accompagnée des informations suivantes :

- **Nom du doctorant** : Affiche le nom complet de chaque doctorant sous la supervision de l'enseignant.
- **Sujet de recherche** : Indique le sujet de recherche spécifique de chaque doctorant, permettant ainsi à l'enseignant de suivre les avancées et les thématiques traitées.
- **État d'avancement** : Optionnellement, l'enseignant peut voir le statut d'avancement de chaque projet de recherche, fournissant une indication sur le progrès des doctorants.

Cette vue d'ensemble est conçue pour faciliter la gestion et le suivi des doctorants, permettant ainsi une interaction plus fluide entre enseignants et étudiants.

Nom du Doctorant	Sujet de Recherche
Essalihi Badia	L'Impact des Microplastiques sur les Écosystèmes Marins et la Chaîne Alimentaire

Le menu latéral gauche de l'application contient les options : Notifications, Documents, Vue globale, Calendar, Evenements et Déconnecter.

FIGURE 39 – Liste des doctorants et leurs sujets de recherche

4.2.2.3 Espace Administrateur

L'espace Administrateur comporte des fonctionnalités similaires à celles des espaces Doctorant et Enseignant, tout en intégrant des sections supplémentaires adaptées aux responsabilités de l'administrateur.

Gestion Admin La section "Gestion Admin" offre une vue globale sur l'ensemble des doctorants, fournissant des informations essentielles telles que :

- **Liste des doctorants** : Affiche tous les doctorants inscrits dans le système avec leurs sujets de recherche , facilitant ainsi la gestion des informations.

Informations des Doctorants					
PRÉNOM	NOM	SUJET DE RECHERCHE	ENSEIGNANT ENCADRANT	DÉPARTEMENT	SPÉCIALISATION
Badiaa	ESsalihI	L'Impact des Microplastiques sur les Écosystèmes Marins et la Chaîne Alimentaire	ESsalihI enseignant2	biologie	biologie
Reda	ESsalihI	sujet	ESsalihI badiaa	info	info

FIGURE 40 – Liste des doctorants et leurs sujets de recherche

- **Répartition des Doctorants par Spécialisation** : Cette section comprend un graphique illustrant la répartition des doctorants par spécialisation.



FIGURE 41 – Répartition des Doctorants par Spécialisation

Demandes Cette section englobe l'ensemble des demandes effectuées par les doctorants, présentées sous forme de tableau. L'administrateur a la possibilité de valider ou de rejeter chaque demande, offrant ainsi une gestion claire et efficace des requêtes. De plus, il peut afficher les détails de chaque demande pour une évaluation approfondie, garantissant ainsi un suivi précis des besoins des doctorants.

Doctorant Requests						
ID Demande	Type Demande	Status	Date de Soumission	Nom	Prénom	Département
157	inscription	en cours de traitement	2024-09-27	ESsalihi	Reda	info
158	retrait-provisoire	en cours de traitement	2024-09-27	ESsalihi	Reda	info
Détails de la Demande Motif de Retrait: Motif de Retrait Date de Début Prévue: 27 septembre 2024 Date de Retour Prévue: 28 septembre 2024						
<button>Valider</button> <button>Rejeter</button>						

FIGURE 42 – Tableau des demandes des doctorants

4.3 Backend

4.3.1 Structure du Projet

La structure du projet backend est conçue pour séparer les responsabilités et rendre le développement plus modulable. Voici un aperçu des principaux dossiers et fichiers du backend :

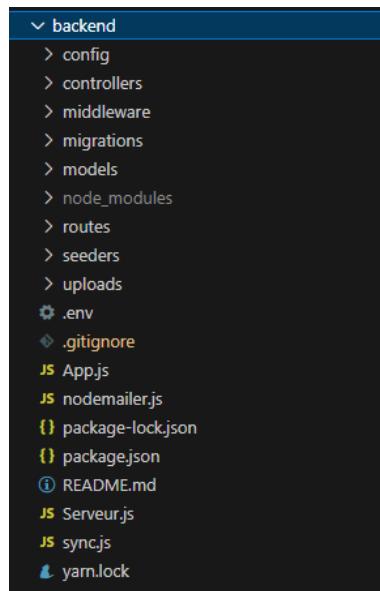


FIGURE 43 – Structure du projet backend.

- **config/** : Contient les fichiers de configuration.
 - **config.js** : Fichier de configuration général.
- **database.js** : Gère la connexion à la base de données (initialisation avec Sequelize ou un autre ORM).
- **controllers/** : Contient les contrôleurs, qui gèrent la logique des requêtes et des réponses pour chaque route.
- **middleware/** : Dossier pour les middlewares, qui sont des fonctions exécutées avant que les requêtes n'atteignent les contrôleurs (authentification, validation, etc.).

- **migrations/** : Contient les fichiers de migration de la base de données (création et modification de tables), généralement gérés avec Sequelize.
- **models/** : Contient les modèles de la base de données, définissant les tables et leurs relations.
- **node_modules/** : Dossier contenant toutes les dépendances et packages installés via npm ou Yarn.
- **routes/** : Définit les routes de l'application, associant chaque route à un contrôleur spécifique.
- **seeders/** : Fichiers pour insérer des données initiales dans la base de données.
- **uploads/** : Dossier où sont stockés les fichiers téléchargés par les utilisateurs (images, documents, etc.).
- **.env** : Contient les variables d'environnement, telles que les clés API et les configurations de base de données.
- **.gitignore** : Liste des fichiers et dossiers à ignorer par Git (comme **node_modules**, **.env**).
- **App.js** : Le fichier principal qui initialise l'application backend, probablement lié à Express.
- **nodemailer.js** : Gère l'envoi d'e-mails avec Nodemailer, une bibliothèque Node.js pour l'envoi d'e-mails.
- **package-lock.json** : Fichier généré automatiquement qui enregistre les versions exactes des packages installés.
- **package.json** : Contient les dépendances du projet et les scripts définis pour les tâches courantes.
- **README.md** : Documentation du projet backend expliquant l'installation et l'utilisation.
- **Serveur.js** : Le fichier serveur, probablement le point d'entrée de l'application, qui démarre le serveur.
- **sync.js** : Fichier utilisé pour synchroniser la base de données, par exemple pour synchroniser les modèles avec la base de données.
- **yarn.lock** : Fichier généré par Yarn qui enregistre les versions exactes des dépendances, assurant la cohérence entre les différentes installations.

4.3.2 Backend test avec Postman

Cette section présente les tests effectués sur le backend à l'aide de Postman pour vérifier les différentes fonctionnalités. Chaque test est illustré par une figure correspondante.

1. **Affichage de tous les utilisateurs** : Test réalisé pour afficher tous les utilisateurs présents dans la base de données.

```

[{"id": "utilisateur_17", "email": "ESSALI.HI.B795@scd.ac.ma", "mot_de_passe_hache": "$2b$10$XCeZXtf7qHiQVOGNx4nWeepsUxqOZwFm4y6McFHqhp2z7UWhaxu", "role": "edudiant", "updatedAt": "2024-07-07T18:23:31.000Z", "createdAt": "2024-07-07T18:23:31.000Z"}, {"id": "utilisateur_33", "email": "essalihbadiaapro@gmail.com", "mot_de_passe_hache": "$2b$10$ZWjyNGKz8VMOMNKLarXgtgGbfmNmVnnsI0870aA4YOCCD4c", "role": "edudiant", "updatedAt": "2024-07-08T21:05:37.000Z", "createdAt": "2024-07-08T21:06:37.000Z"}, {"id": "utilisateur_34", "email": "edudiant34@gmail.com", "mot_de_passe_hache": "$2b$10$IA0ahqaixH3HHtpPHGeDO/HgYieb7ykZJQicPuEEIUCljgvw6", "role": "edudiant", "prenom": "Edouard", "updatedAt": "2024-07-08T21:22:41.000Z", "createdAt": "2024-07-08T21:22:41.000Z"}, {"id": "utilisateur_35", "email": "redessa2011@gmail.com", "mot_de_passe_hache": "$2b$10$VuHi4Pen2mc0fmsPe5K4t.Fv7q2lhegrzHnfqKsxErg8zwoE/Gm", "role": "edudiant", "prenom": "Redessa", "updatedAt": "2024-07-08T21:27:12.000Z", "createdAt": "2024-07-08T21:27:12.000Z"}, {"id": "utilisateur_38", "email": "redessa2011@gmail.com", "mot_de_passe_hache": "$2b$10$Cj.SkFSEqJNP0wh3.o4.siqOAGKqNnSrVzv2j7QmXG9mZtCtsa", "role": "edudiant", "prenom": "Redessa", "updatedAt": "2024-07-08T22:05:40.000Z", "createdAt": "2024-07-08T22:05:40.000Z"}]

```

FIGURE 44 – Test d'affichage de tous les utilisateurs.

2. **Test de login** : Vérification de la fonctionnalité de connexion pour s'assurer que les utilisateurs peuvent se connecter avec des identifiants valides.

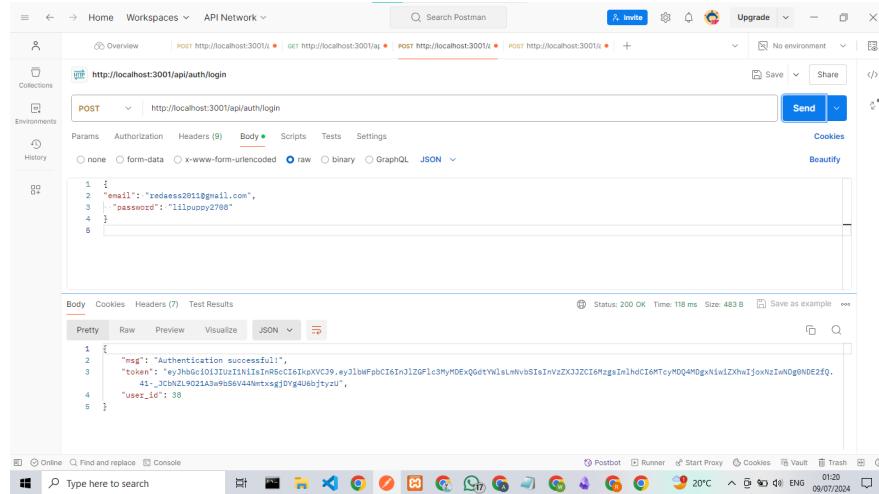


FIGURE 45 – Test de la fonctionnalité de login.

3. **Inscription** : Test de la fonctionnalité d'inscription pour valider le processus d'ajout d'un nouvel utilisateur.

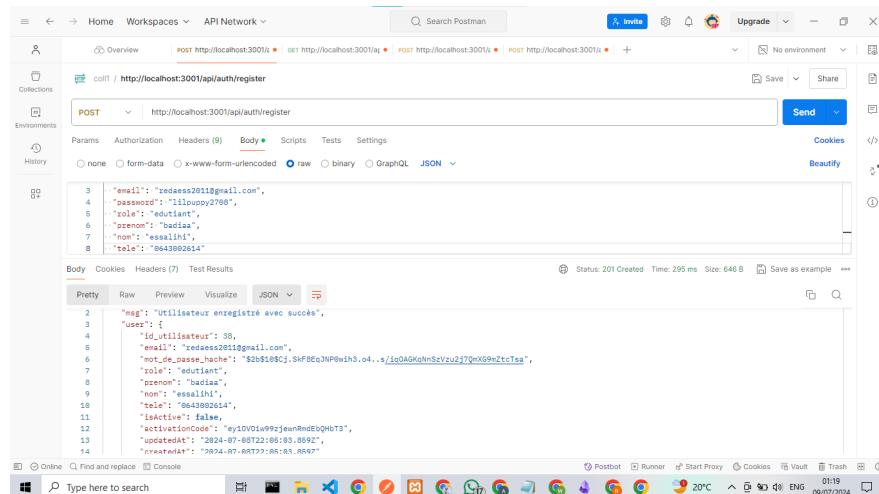


FIGURE 46 – Test de la fonctionnalité d'inscription.

4. **Réinitialisation du mot de passe** : Test de la fonctionnalité de réinitialisation du mot de passe pour garantir son bon fonctionnement.

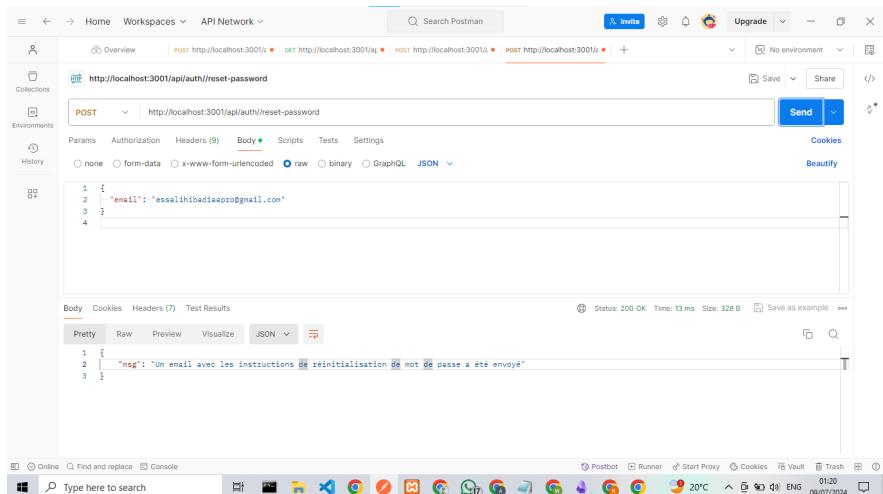


FIGURE 47 – Test de réinitialisation du mot de passe.

5. Insertion des informations : Test réalisé pour insérer des informations dans la base de données.

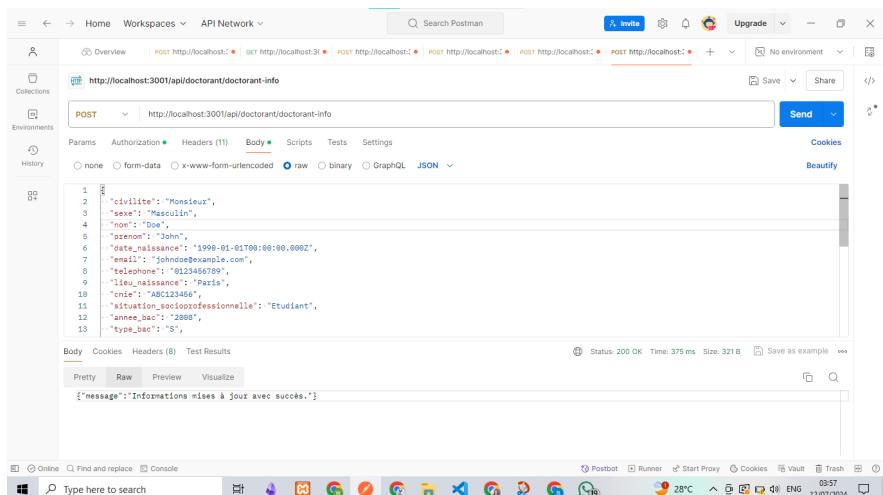


FIGURE 48 – Test d'insertion des informations.

5 Conclusion

Ce projet d'application web pour la gestion du service recherche représente une avancée significative dans l'amélioration des interactions entre doctorants, enseignants et administrateurs. En centralisant les informations et en facilitant la communication, l'application vise à optimiser la gestion des projets de recherche, des thèses et des activités académiques. Grâce à des fonctionnalités adaptées aux besoins spécifiques de chaque utilisateur, l'application favorisera une meilleure organisation et un suivi efficace des démarches administratives.

La conception d'une interface intuitive et responsive garantit une accessibilité accrue, répondant ainsi aux exigences contemporaines en matière de technologie éducative. En intégrant des mesures de sécurité robustes et une planification méthodique, ce projet est conçu pour être pérenne et évolutif, prêt à s'adapter aux futures exigences et défis du secteur académique.

En définitive, la réalisation de cette application contribuera non seulement à l'efficacité de la gestion académique, mais également à l'enrichissement de l'expérience des utilisateurs, rendant ainsi l'environnement de recherche plus collaboratif et dynamique. Ce projet est une étape vers l'optimisation des processus académiques et une réponse aux attentes des acteurs impliqués dans le monde de la recherche.

6 Références

- <https://nodejs.org/en/docs/>
- <https://expressjs.com/en/4x/api.html>
- <https://sequelize.org/master/>
- <https://www.postgresql.org/docs/>
- <https://reactjs.org/docs/getting-started.html>
- <https://redux.js.org/introduction/getting-started>
- <https://mui.com/getting-started/installation/>
- <https://chakra-ui.com/docs/getting-started>
- <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- <https://tailwindcss.com/docs/installation>
- <https://axios-http.com/docs/intro>
- <https://reactrouter.com/en/main>
- <https://formik.org/docs/overview>
- <https://www.chartjs.org/docs/latest/>
- <https://github.com/jquense/react-big-calendar>
- <https://nodemailer.com/about/>
- <https://www.phpmyadmin.net/docs/>