RAPPORT DU PFE

**Sujet du PFE : Création d’une application web pour la gestion de départements**

**En utilisant (MERN\* Stack) technologie.**

MERN\* : MongoDB-Express-ReactJS-NodeJS

Période du projet : DD MM 2022 – DD MM 2022

Réalisépar:

**Ouadie Zerhouni**

**Abdessamad Belmadani**

Encadré par :

**Mr.Mehdi Tmimi**

# Sommaire

Contents

[Sommaire 2](#_Toc131300435)

[Remerciement 5](#_Toc131300436)

[Table d’abréviation 6](#_Toc131300437)

[MERN 6](#_Toc131300438)

[MongoDB, Express.js, React.js, Node.js 6](#_Toc131300439)

[HTML 6](#_Toc131300440)

[HyperText Markup Language 6](#_Toc131300441)

[JS 6](#_Toc131300442)

[JavaScript 6](#_Toc131300443)

[CSS 6](#_Toc131300444)

[Cascading Style Sheets 6](#_Toc131300445)

[ORM 6](#_Toc131300446)

[Object-Relational Mapping 6](#_Toc131300447)

[API 6](#_Toc131300448)

[Application Programming Interface 6](#_Toc131300449)

[REST API 6](#_Toc131300450)

[Representational State Transfer API 6](#_Toc131300451)

[HTTP 6](#_Toc131300452)

[Hypertext Transfer Protocol 6](#_Toc131300453)

[JSON 6](#_Toc131300454)

[JavaScript Object Notation 6](#_Toc131300455)

[SQL 6](#_Toc131300456)

[Structured Query Language 6](#_Toc131300457)

[NoSQL 6](#_Toc131300458)

[Not Only SQL 6](#_Toc131300459)

[URL 6](#_Toc131300460)

[Uniform Resource Locator 6](#_Toc131300461)

[JWT 6](#_Toc131300462)

[JSON Web Token 6](#_Toc131300463)

[UML 6](#_Toc131300464)

[Unified Modeling Language 6](#_Toc131300465)

[INTRODUCTION 7](#_Toc131300466)

[Chapitre 1 : Cadre et contexte du projet 8](#_Toc131300467)

[Introduction 8](#_Toc131300468)

[Enjeux et contexte du projet 8](#_Toc131300469)

[Présentation de l’organisme d’accueil 9](#_Toc131300470)

[Chapitre 2 : Présentation du projet et Cahier de charges 9](#_Toc131300471)

[Introduction : 9](#_Toc131300472)

[Présentation du projet : 9](#_Toc131300473)

[Fonctionnalités : 10](#_Toc131300474)

[Contenu de l’application 14](#_Toc131300475)

[Arborescence de l’application (todo) 15](#_Toc131300476)

[Planification du projet(todo) 15](#_Toc131300477)

[Chapitre 3 : Technologies et outils utilisés 16](#_Toc131300478)

[Introduction 16](#_Toc131300479)

[Technologie implémentées 16](#_Toc131300480)

[MongoDB 16](#_Toc131300481)

[Express.js 17](#_Toc131300482)

[\*RESTful api : 18](#_Toc131300483)

[React.js 19](#_Toc131300484)

[Node.js 19](#_Toc131300485)

[JWT 20](#_Toc131300486)

[Outils d’implémentation 20](#_Toc131300487)

[Visual Studio Code : 20](#_Toc131300488)

[Git 21](#_Toc131300489)

[GitHub 22](#_Toc131300490)

[Studio 3t 23](#_Toc131300491)

[Chapitre 4 : Architecture et conception : 24](#_Toc131300492)

[Introduction 24](#_Toc131300493)

[Architecture 3 Tiers 24](#_Toc131300494)

[C’est quoi une architecture 3 tiers : 24](#_Toc131300495)

[Dans notre cas : 25](#_Toc131300496)

[Conception de l’application : 27](#_Toc131300497)

# Remerciement

Nous tenons à remercier vivement notre encadrant du PFE Mr MEHDI TMIMIpour leur disponibilité, leur écoute et leurs conseils avisés.

Nous tenons à remercier aussi toutes les personnes qui ont contribué au succès de notre PFE, qui nous ’ont accompagné et aidé lors de la rédaction de ce rapport.

Persons.foreach(personne=>{

console.log(‘Merci ’+ personne)

})

# Table d’abréviation

|  |  |
| --- | --- |
| **MERN** | MongoDB, Express.js, React.js, Node.js |
| **HTML** | HyperText Markup Language |
| **JS** | JavaScript |
| **CSS** | Cascading Style Sheets |
| **ORM** | Object-Relational Mapping |
| **API** | Application Programming Interface |
| **REST API** | Representational State Transfer API |
| **HTTP** | Hypertext Transfer Protocol |
| **JSON** | JavaScript Object Notation |
| **SQL** | Structured Query Language |
| **NoSQL** | Not Only SQL |
| **URL** | Uniform Resource Locator |
| **JWT** | JSON Web Token |
| **UML** | Unified Modeling Language |

# INTRODUCTION

Le rapport de fin d'étude représente un aspect crucial de la formation académique d'un étudiant. Il permet de démontrer les compétences acquises au cours de ses années d'études et de les appliquer à un projet concret. En effet, le projet de fin d'étude est une opportunité unique de mettre en pratique les connaissances acquises au fil des années pour produire une solution innovante à un problème réel.

La réalisation d'un projet de fin d'étude ajoute également une valeur considérable à la formation académique d'un étudiant. Cela lui permet d'acquérir une expérience concrète en matière de résolution de problèmes, de travail en équipe et de présentation de résultats. De plus, ce projet peut également donner lieu à des publications scientifiques ou à des opportunités de poursuite de carrière.

En somme, le projet de fin d'étude est une occasion unique pour les étudiants de démontrer leurs compétences et leur capacité à apporter une contribution significative à leur domaine d'études. Ce rapport sera l'occasion de décrire les étapes de ce projet et les résultats obtenus.

# Chapitre 1 : Cadre et contexte du projet

## Introduction

Ce premier chapitre est consacré à la mise en avant du contexte du projet, qui a été réalisé dans le cadre du projet de fin d'étude. Il présentera également l'organisme qui nous a accompagnés tout au long de cette expérience,

## Enjeux et contexte du projet

Notre projet de fin d'étude porte sur la conception et la mise en place d'une application web destinée à la gestion de départements.

Pour ce faire, nous avons choisi d'utiliser la technologie MERN Stack, qui est une combinaison de plusieurs outils et frameworks tels que MongoDB, Express.js, React.js et Node.js.

Cette technologie nous permettra de développer une application web performante et évolutive, en utilisant des outils adaptés à chaque étape de la création et de la gestion de l'application.

Nous sommes convaincus que cette approche nous permettra de répondre de manière optimale aux besoins de notre public cible et de garantir une expérience utilisateur fluide et satisfaisante.

## Présentation de l’organisme d’accueil

**USMBA :** Université Sidi Mohamed Ben Abdellah

**EST-Fès :** Ecole Supérieure de Technologie Fès

# Chapitre 2 : Présentation du projet et Cahier de charges

## Introduction :

L'objectif de ce chapitre consiste à réaliser une analyse fonctionnelle approfondie de l'application en examinant ses différents aspects. Nous commencerons par explorer les raisons qui ont conduit à sa création, puis nous exposerons les objectifs fonctionnels du projet. Par la suite, nous aborderons le public cible visé par l'application et présenterons de manière détaillée l'ensemble des fonctionnalités ainsi que les exigences en matière de charte graphique.

## Présentation du projet :

Notre objectif est de créer un service web complet dédié à la gestion des départements, qui sera accessible aux visiteurs standards, aux professeurs et aux chefs de département. Ce service offrira à tous les utilisateurs la possibilité de consulter des informations sur les départements, telles que leur organisation interne, leurs programmes et leurs événements. En outre, les utilisateurs pourront accéder aux emplois du temps pour connaître les horaires de cours et de rendez-vous.

En plus de ces fonctionnalités générales, les utilisateurs(professeurs) pourront également laisser des commentaires sur les procès-verbaux ou les décisions prises lors des réunions de département. Cette fonctionnalité permettra une transparence accrue et facilitera la communication entre les membres du département.

Les chefs de département bénéficieront de fonctionnalités supplémentaires, qui leur permettront de gérer leur département de manière plus efficace. Ils auront ainsi accès à des outils de planification avancée pour la gestion des emplois du temps et pourront attribuer des tâches spécifiques à chaque membre du département en fonction de leurs compétences et de leurs disponibilités. En somme, notre service web offrira une solution complète et flexible pour la gestion des départements, en répondant aux besoins des différents types d'utilisateurs.

## Fonctionnalités :

#### Fonctionnalités pour les visiteurs standards :

* Consulter la liste des départements disponibles
* Afficher les informations relatives à chaque département (nom, description, liste des professeurs, etc.)
* Consulter les emplois du temps des départements

Voir figure 1 au-dessous :

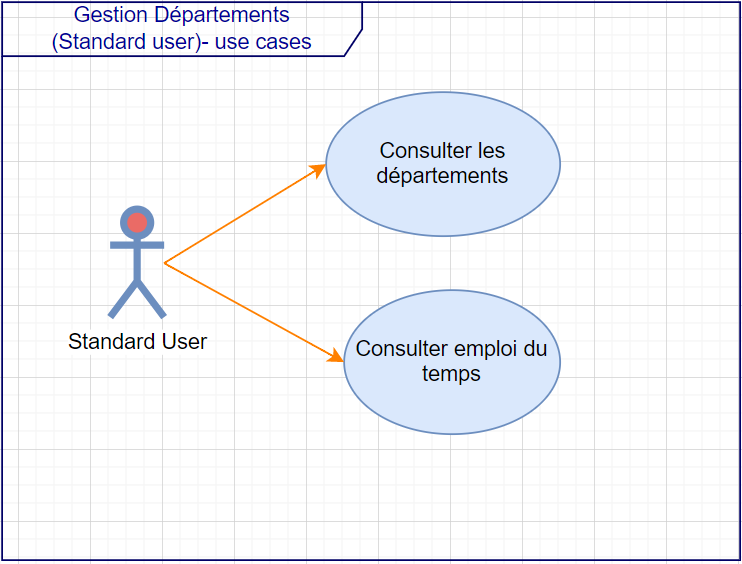


Figure 1

#### Fonctionnalités pour les professeurs :

* Consulter la liste des départements disponibles
* Afficher les informations relatives à chaque département (nom, description, liste des professeurs, etc.)
* Consulter les emplois du temps des départements
* Consulter et laisser des commentaires sur les PV

Voir figure 2 au-dessous :

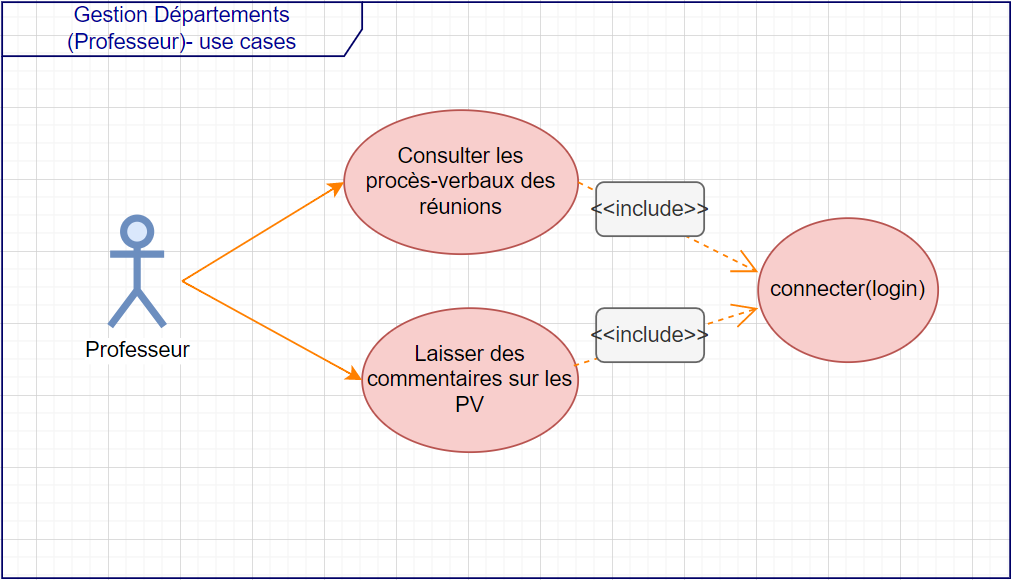


Figure 2

#### Fonctionnalités pour les chefs de département :

* Consulter la liste des départements disponibles
* Afficher les informations relatives à chaque département (nom, description, liste des professeurs, etc.)
* Consulter les emplois du temps des départements
* Laisser des commentaires sur les PV
* Gérer les professeurs de son département (ajouter, modifier, supprimer)
* Gérer les filières et les options de son département (ajouter, modifier, supprimer)
* Gérer les emplois du temps de son département (ajouter, modifier, supprimer)

Voir figure 3 au-dessous :

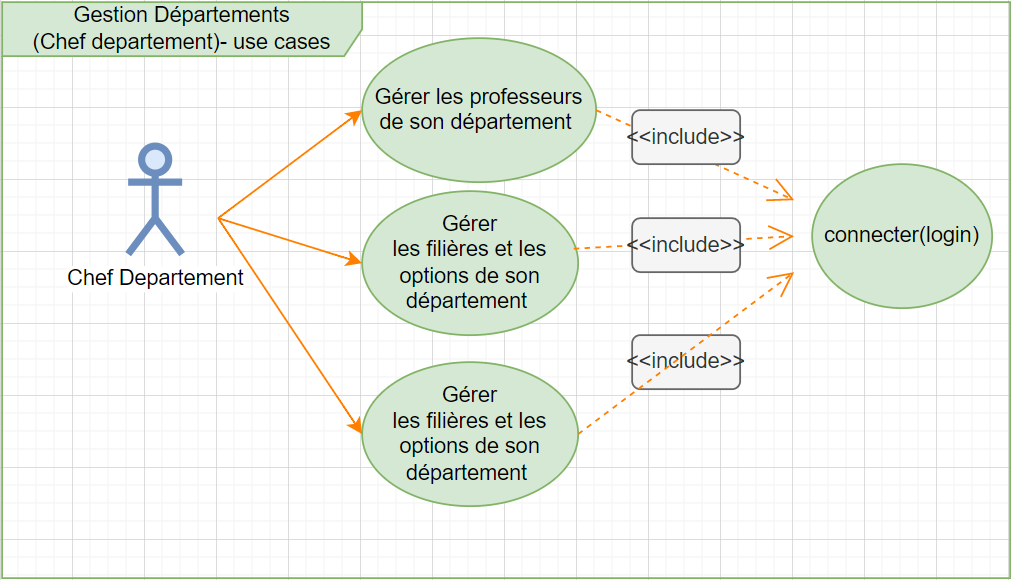


Figure 3

#### Fonctionnalités pour l’Admin :

* Consulter la liste des départements disponibles
* Afficher les informations relatives à chaque département (nom, description, liste des professeurs, etc.)
* Consulter les emplois du temps des départements
* Laisser des commentaires sur les PV
* Gérer les professeurs (ajouter, modifier, supprimer)
* Gérer les filières, les options et les départements (ajouter, modifier, supprimer)
* Gérer les emplois du temps des départements (ajouter, modifier, supprimer)

Voir figure 4 au-dessous :

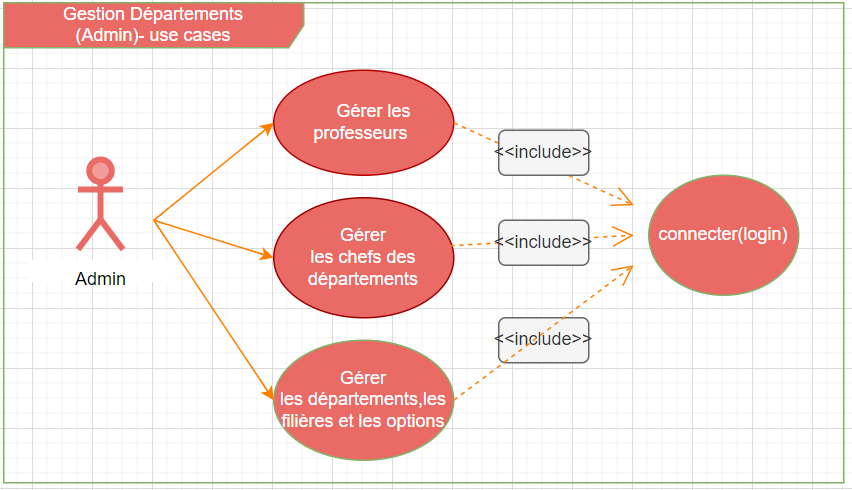


Figure 4

## Contenu de l’application

Notre application web est destinée à fournir un service complet pour la gestion des départements. Elle sera accessible aux visiteurs standards, aux professeurs et aux chefs de département. L'objectif est d'offrir à tous les utilisateurs la possibilité de consulter des informations sur les départements, y compris leur organisation interne, leurs programmes et leurs événements, ainsi que d'accéder aux emplois du temps pour connaître les horaires de cours et de rendez-vous.

En outre, les professeurs pourront également utiliser l'application pour laisser des commentaires sur les procès-verbaux ou les décisions prises lors des réunions de département. Cette fonctionnalité permettra d'augmenter la transparence et de faciliter la communication entre les membres du département.

Les chefs de département auront accès à des fonctionnalités supplémentaires pour gérer leur département de manière plus efficace. Ils pourront ainsi utiliser des outils de planification avancée pour gérer les emplois du temps et attribuer des tâches spécifiques à chaque membre du département en fonction de leurs compétences et de leurs disponibilités. En somme, notre application web offrira une solution complète et flexible pour la gestion des départements, répondant ainsi aux besoins des différents types d'utilisateurs, y compris les visiteurs standards, les professeurs et les chefs de département.

Charte graphique : exigences et limites

L'application web que nous développons a un caractère académique et doit donc respecter les caractéristiques propres à un tel site web. Nous cherchons à fournir une interface utilisateur à la fois professionnelle et conviviale pour les professeurs et les étudiants. La charte graphique de l'application se doit d'être simple et facile à utiliser, tout en restant professionnelle.

Nous avons également pris en compte les couleurs de l'Université Sidi Mohamed Ben Abdellah pour la conception de la charte graphique de l'application. Ainsi, les couleurs principales du site web seront le rouge et le bleu, qui représentent les couleurs de l'université. Nous veillerons à ce que la charte graphique de l'application soit en adéquation avec les exigences et les limites imposées par le contexte académique dans lequel elle sera utilisée.

## Arborescence de l’application (todo)

## Planification du projet(todo)

# Chapitre 3 : Technologies et outils utilisés

## Introduction

Ce projet consiste en une application web développée à l'aide d'un ensemble de technologies appelé MERN stack. Cette dernière est composée de plusieurs outils de développement tels que MongoDB, Express.js, React.js et Node.js.

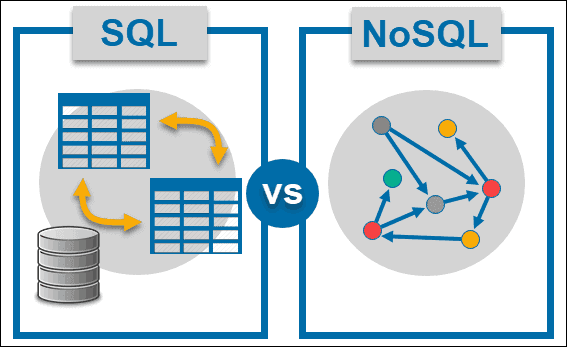
## Technologie implémentées

L'application web en question a été développée à l'aide d'un ensemble de technologies appelé MERN stack. Cette dernière est une suite d'outils de développement qui combine quatre technologies majeures : **MongoDB**, **Express.js**, **React.js** et **Node.js**.

### MongoDB

Commençons par MongoDB, qui est un système de gestion de base de données NoSQL adaptée au stockage de données non structurées, semi-structurées ou mal structurées. Contrairement aux bases de données relationnelles traditionnelles,

MongoDB permet de stocker des données sous forme de documents JSON, offrant ainsi une plus grande flexibilité pour stocker et manipuler les données. En outre, MongoDB offre une grande évolutivité horizontale et permet de distribuer les données sur plusieurs serveurs pour améliorer la performance et la disponibilité de la base de données. La réplication de données est également utilisée pour assurer la haute disponibilité des données en cas de défaillance d'un serveur.



### Express.js

Express.js est une bibliothèque de développement côté serveur pour Node.js. Elle permet de créer facilement des serveurs web et des applications RESTful grâce à un ensemble de fonctionnalités utiles. La plateforme facilite la gestion des requêtes et des réponses HTTP, ainsi que l'écriture de middleware pour intercepter les requêtes entrantes et sortantes.



L'un des principaux avantages d'Express.js est sa flexibilité. Elle permet aux développeurs de choisir leur propre structure d'application et de l'adapter en fonction de leurs besoins. La plateforme permet également une intégration facile avec des bibliothèques tierces telles que des bases de données, des outils d'authentification et des moteurs de modèles.

Express.js offre également une performance élevée en termes de vitesse de traitement des requêtes HTTP, ce qui permet de traiter efficacement un grand nombre de demandes simultanées. La plateforme permet également une optimisation facile du cache pour améliorer les temps de réponse et la gestion des ressources.

Enfin, l'utilisation d'Express.js facilite le développement collaboratif grâce à sa documentation complète et à une large communauté de développeurs. Les développeurs peuvent facilement trouver des exemples de code et des réponses à des questions courantes sur les forums en ligne.

En résumé, Express.js est une plateforme de développement de logiciels côté serveur pour Node.js qui offre une grande flexibilité, une performance élevée, une optimisation facile du cache et une communauté de développeurs active. Elle est un choix judicieux pour les développeurs qui cherchent à créer facilement des serveurs web et des applications RESTful\*.

### \*RESTful api :

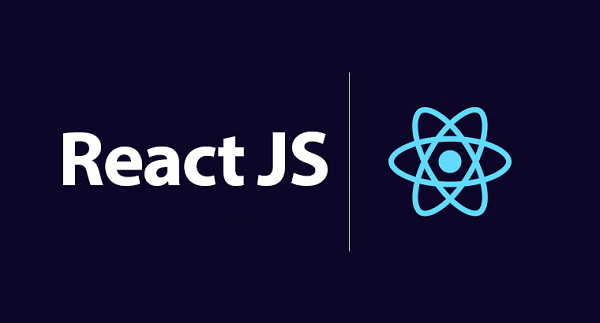
L'API RESTful, également appelée "Representational State Transfer" en anglais, est un style d'architecture pour les applications web qui utilise les protocoles et les méthodes HTTP pour la transmission de données. Grâce à cette architecture, les clients et les serveurs peuvent échanger des données de manière simple, cohérente et efficace en utilisant des ressources identifiées par des URL et en échangeant des représentations de ces ressources sous forme de messages.

En pratique, l'API RESTful fournit une interface permettant à des applications de communiquer entre elles à l'aide de requêtes HTTP standard telles que GET, POST, PUT et DELETE. Les données sont échangées sous forme de représentations de ressources généralement encodées en format JSON ou XML. Les URI (Uniform Resource Identifier) sont utilisées pour identifier les ressources pouvant être manipulées via l'API.

Les avantages de l'API RESTful sont multiples. Tout d'abord, elle offre une grande flexibilité dans la manipulation des données, ce qui facilite leur utilisation et leur mise à jour. En outre, cette architecture est largement utilisée et bien documentée, ce qui facilite son intégration dans d'autres applications. Enfin, l'utilisation de protocoles standard tels que HTTP permet une communication efficace et cohérente entre les applications.

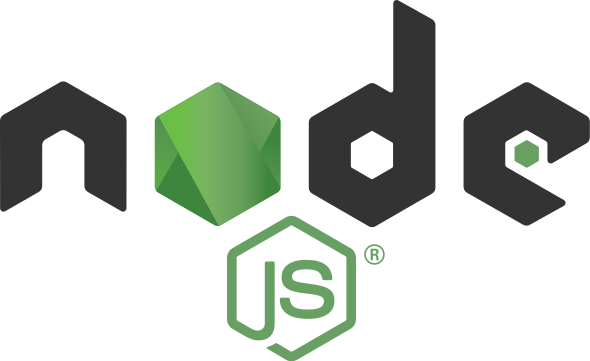
# React.js

React.js, quant à lui, est une bibliothèque JavaScript pour la création d'interfaces utilisateur. Elle est utilisée pour construire des composants réutilisables qui peuvent être combinés pour former des interfaces utilisateur plus complexes. React.js utilise une approche de programmation déclarative qui permet de décrire comment l'interface utilisateur doit être rendue en fonction de l'état actuel de l'application. Cette approche permet de simplifier la création d'interfaces utilisateur interactives et dynamiques.



### Node.js

Enfin, Node.js est un environnement d'exécution JavaScript côté serveur. Il permet d'exécuter du code JavaScript en dehors d'un navigateur web, ce qui ouvre de nombreuses possibilités pour le développement d'applications web. Node.js est souvent utilisé pour créer des serveurs web, des applications en temps réel, des outils de ligne de commande, etc.



En combinant ces quatre technologies, la MERN stack offre un ensemble complet d'outils pour le développement d'applications web modernes. Elle permet aux développeurs de créer des applications évolutives, performantes et réactives, tout en réduisant les coûts de développement et en accélérant les délais de mise sur le marché. La popularité de la MERN stack ne cesse de croître dans la communauté des développeurs, et elle est devenue une technologie de choix pour de nombreuses entreprises qui cherchent à construire des applications web modernes et robustes.

### JWT

JWT (JSON Web Token) est un format de jeton d'authentification, qui est utilisé pour sécuriser les échanges entre des systèmes distribués tels que les applications web ou les services web.



Le format JWT est basé sur des objets JSON (JavaScript Object Notation) qui contiennent des informations d'identification et des données supplémentaires telles que des informations d'expiration, des revendications (claims) et des autorisations. Les données contenues dans le jeton peuvent être encodées et signées pour assurer l'intégrité et la confidentialité des données échangées.

Un jeton JWT est émis par un serveur d'authentification après une demande d'authentification réussie. Il peut être stocké localement sur un client ou envoyé dans une requête HTTP. Lorsque le jeton est reçu par un serveur, il peut être vérifié pour assurer que le jeton n'a pas été altéré et qu'il est encore valide.

Les JWT sont largement utilisés pour l'authentification et l'autorisation dans les applications web modernes, notamment pour l'authentification d'API REST. Les avantages de JWT sont leur portabilité, leur sécurité, leur légèreté et leur facilité d'utilisation.

## Outils d’implémentation

### Visual Studio Code :

VS Code (ou Visual Studio Code) est un éditeur de code source développé par Microsoft pour les systèmes d'exploitation Windows, Linux et macOS. Il s'agit d'un outil très populaire parmi les développeurs pour écrire, déboguer et gérer du code dans une grande variété de langages de programmation.



VS Code est une application légère et rapide qui offre des fonctionnalités avancées telles que la coloration syntaxique, l'auto-complétion, la mise en forme de code, le débogage interactif, le contrôle de version et l'intégration avec d'autres outils de développement tels que Git, GitHub, Docker, etc.

De plus, VS Code est extensible via un écosystème de plugins qui permettent d'ajouter des fonctionnalités supplémentaires à l'éditeur pour répondre aux besoins spécifiques des développeurs. Les utilisateurs peuvent également personnaliser l'interface utilisateur, les thèmes et les raccourcis clavier pour une expérience de développement plus personnalisée.

### Git

Git est un système de contrôle de version de logiciel open source créé par Linus Torvalds en 2005. Il est utilisé pour suivre les modifications apportées au code source d'un projet et pour collaborer sur ce code source entre plusieurs développeurs.



Avec Git, les développeurs peuvent créer des "commits" (ou "instantanés") du code source de leur projet à des moments donnés, créant ainsi une série chronologique d'instantanés qui peut être utilisée pour suivre les modifications, revenir à des versions précédentes ou résoudre des conflits entre les modifications apportées par plusieurs développeurs.

Git permet également de travailler en équipe sur un même projet en utilisant des "branches" qui permettent de créer une copie du code source pour travailler sur une nouvelle fonctionnalité ou une nouvelle correction de bug, sans perturber la version principale du code. Les développeurs peuvent ensuite fusionner leur travail dans la version principale du code source lorsqu'ils ont terminé.

Git est un outil très puissant pour la gestion de versions de code source, utilisé par de nombreuses entreprises et projets open source pour suivre les modifications de leur code source et collaborer efficacement sur leur développement.

### GitHub

GitHub est une plateforme de développement de logiciels qui permet aux développeurs de collaborer et de partager leur travail. Elle offre un système de contrôle de version distribué pour la gestion de projets de développement de logiciels, qui permet de suivre les modifications apportées au code source et de gérer les versions multiples d'un projet.



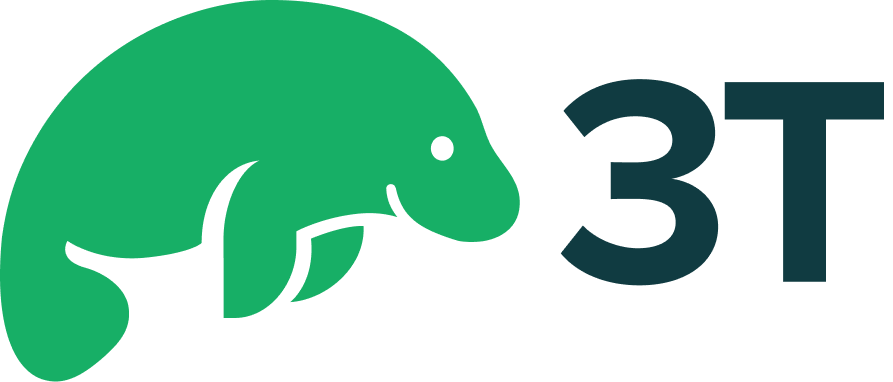
Sur GitHub, les développeurs peuvent héberger leur code source et collaborer avec d'autres développeurs pour créer des projets de logiciels open source. Les utilisateurs peuvent également créer des projets privés pour une utilisation interne au sein d'une organisation ou d'une entreprise.

En plus de la gestion de version, GitHub propose des outils de collaboration tels que les demandes de tirages (pull requests) et les problèmes (issues) pour faciliter la communication entre les membres de l'équipe. Elle offre également une intégration avec des outils de développement tiers tels que les intégrations continues (CI) et les déploiements continus (CD).

En somme, GitHub est une plateforme essentielle pour les développeurs qui cherchent à collaborer et à partager leur travail, et qui veulent suivre et gérer les versions de leur code source.

### Studio 3t

Studio 3T est un outil de gestion de bases de données MongoDB pour les développeurs, les ingénieurs de données et les administrateurs de bases de données. Il fournit une interface graphique conviviale pour interagir avec des bases de données MongoDB, ainsi que des fonctionnalités avancées pour la manipulation et la visualisation des données.



Studio 3T prend en charge la création, la modification et la suppression de bases de données et de collections, ainsi que la recherche, l'exportation et l'importation de données. Il permet également de créer des requêtes MongoDB avancées en utilisant la syntaxe de requête MongoDB standard, ainsi que de visualiser et de modifier les résultats de ces requêtes.

En outre, Studio 3T fournit des fonctionnalités de gestion des index, de surveillance en temps réel des opérations de base de données, de sauvegarde et de restauration de bases de données, ainsi que des fonctionnalités pour la génération de code pour les langages de programmation couramment utilisés.

En somme, Studio 3T est un outil complet et pratique pour la gestion de bases de données MongoDB, qui permet de simplifier les tâches de développement, de maintenance et de gestion de bases de données.

# Chapitre 4 : Architecture et conception :

## Introduction

Dans ce chapitre on va parler sur l’architecture opter pour concevoir notre application (Architecture 3-Tiers, Langage UML)

## Architecture 3 Tiers

### C’est quoi une architecture 3 tiers :

L'architecture 3 tiers est une approche de développement logiciel qui sépare une application en trois couches ou niveaux distincts : la couche de présentation (ou interface utilisateur), la couche métier (ou logique de traitement) et la couche de données (ou stockage).

Couche de présentation :

La couche de présentation est la couche supérieure de l'architecture 3 tiers. Elle est responsable de l'interaction avec l'utilisateur et fournit l'interface utilisateur de l'application. Cette couche est généralement développée à l'aide de langages de programmation pour le web tels que HTML, CSS, JavaScript ou des frameworks comme ReactJS ou Angular. Elle peut également comprendre des composants d'interface utilisateur comme des boutons, des formulaires et des graphiques.

Couche métier :

La couche métier est la couche intermédiaire de l'architecture 3 tiers. Elle gère la logique de traitement et la manipulation des données. Cette couche est responsable de l'exécution des opérations et des calculs sur les données entrées par l'utilisateur et les données stockées dans la couche de données. Elle peut également comprendre des composants tels que des services, des contrôleurs et des gestionnaires de transactions.

Couche de données :

La couche de données est la couche inférieure de l'architecture 3 tiers. Elle est responsable de la gestion des données et de leur persistance. Cette couche stocke les données dans des bases de données relationnelles ou non relationnelles. Elle peut également comprendre des composants tels que des ORM (Object-Relational Mapping) ou des API d'accès aux données.

L'architecture 3 tiers permet une séparation claire des responsabilités entre les différentes couches, ce qui rend l'application plus modulaire et plus facile à maintenir. Elle permet également une meilleure évolutivité de l'application, car chaque couche peut être développée, testée et déployée indépendamment. En outre, cette architecture facilite l'intégration de nouveaux composants et de nouvelles fonctionnalités, car les différentes couches peuvent être mises à jour sans affecter les autres.



Figure 5 : Architecture 3 tiers

### Dans notre cas :

Dans le cas de notre projet de gestion des départements, l'architecture 3 tiers sera particulièrement utile pour séparer les différents types de fonctionnalités qu’elle offre l’application, et pour permettre une évolutivité facile. La couche de présentation sera représentée par une interface utilisateur développée en **ReactJS**, la couche métier sera gérée par votre backend en utilisant **ExpressJS** et **MongoDB** pour la gestion des données, tandis que la couche de stockage de données sera représentée par la base de données **MongoDB**.

En utilisant une architecture 3 tiers, nous pouvons facilement ajouter de nouvelles fonctionnalités à notre application en ajoutant simplement une nouvelle couche. Par exemple, nous pouvons ajouter une couche supplémentaire pour la gestion des étudiants, sans avoir à modifier la couche de présentation ou la couche métier.

En conclusion, l'architecture 3 tiers est une approche de développement logiciel qui offre de nombreux avantages, notamment la modularité, la réutilisabilité et la facilité de maintenance. Dans le cadre de notre projet de gestion des départements, cette architecture nous permettra de séparer les différentes fonctionnalités de manière efficace, de faciliter l'ajout de nouvelles fonctionnalités et d'assurer une évolutivité facile.

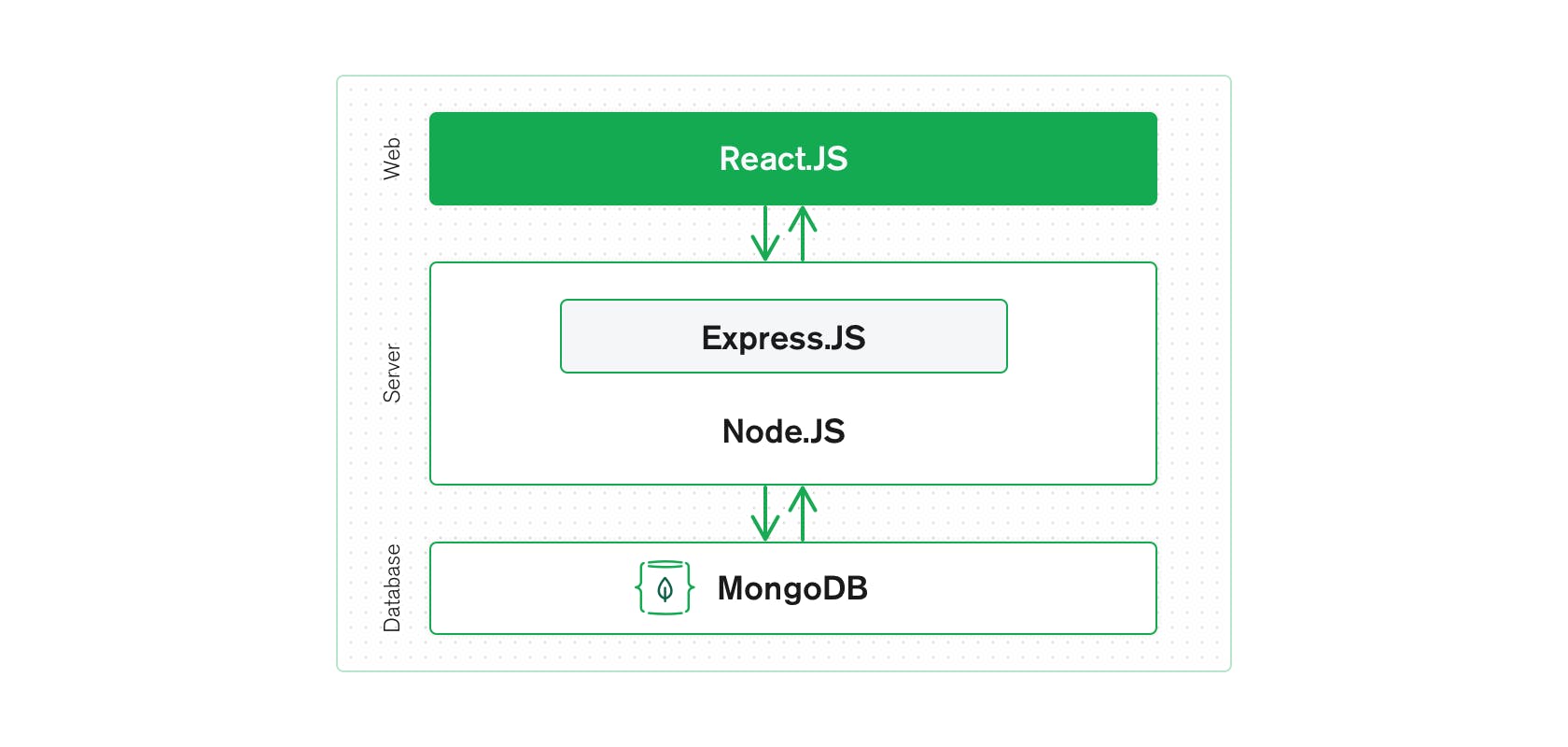
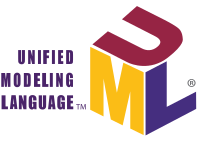


Figure 6 : Architecture 3 tiers MERN

## Conception de l’application :

Pour la conception de notre application nous avons opter pour UML (Unified Modeling Language).



Qui est un langage de modélisation graphique standardisé, utilisé pour spécifier, visualiser, concevoir et documenter les systèmes logiciels et les processus métier.

UML a été créé pour faciliter la communication entre les parties prenantes impliquées dans la conception et la mise en œuvre de systèmes d'information complexes. Il offre une notation graphique standardisée pour représenter les différents aspects d'un système, tels que la structure, le comportement, les interactions, les relations entre les éléments, etc.

Les diagrammes UML les plus couramment utilisés sont :

* Les diagrammes de cas d'utilisation : ils décrivent les interactions entre les utilisateurs et le système, en identifiant les différentes actions que les utilisateurs peuvent effectuer et les résultats attendus.
* Les diagrammes de classes : ils montrent les différentes classes du système, leurs attributs et leurs relations, ainsi que les méthodes et les opérations qu'elles peuvent effectuer.
* Les diagrammes de séquence : ils montrent l'ordre chronologique des interactions entre les objets du système, en mettant en évidence les messages échangés entre eux.
* Les diagrammes d'activité : ils montrent le flux de contrôle entre les activités du système, en montrant les étapes et les décisions prises dans un processus.
* UML est largement utilisé dans le domaine de l'ingénierie logicielle, mais il peut également être utilisé pour modéliser des processus métier, des systèmes embarqués, des systèmes d'information géographique, etc.