

# PLATEFORME E-LEARNING

 VirtualClass

## Encadré par:

- Pr . Abdelhay Haqiq

## Réalisé par:

- |                   |                  |
|-------------------|------------------|
| • Benjeddi Ikrame | • Taleb Lamya    |
| • Kalim Amal      | • Zariouh Salima |

# **Remerciements**

Nous souhaitons dans un premier temps remercier, notre encadrant Monsieur Mr. Abdihay Haqiq, professeur à l'institut national des Postes et Télécommunications, pour son accompagnement tout au long de la période du projet et son partage de ses connaissances en matière de développement web, ainsi que pour sa proposition de projet qui nous a permis d'acquérir des nouvelles compétences.

Nous remercions également le corps professoral et administratif de l'Institut National des Postes et Télécommunications pour la qualité de l'enseignement offert et leur soutien.

# Résumé

Ce présent rapport est rédigé dans le cadre du projet de fin de matières "Ingénierie des applications Web moderne", faisant partie de la formation du cycle d'ingénieur de l'Institut National des Postes et Télécommunications.

Ce projet consiste à concevoir et développer une plateforme E-learning qui offre au professeur un espace de design de cours personnalisé comme valeur ajoutée, ainsi que l'ensemble des fonctionnalités de base d'une application E-learning .

Afin d'atteindre cet objectif, on a eu recours au différentes technologies principalement Node.js pour le backend et React.js pour le frontend.

A travers ce rapport , nous montrons en détail les différentes étapes prises pour la réalisation de ce projet .

# **Abstract**

This report is written within the framework of the project of end of subject "Engineering of the modern Web applications", forming part of the training of the cycle of engineer of the National Institute of the Posts and Telecommunications.

This project consists in designing and developing an E-learning platform that offers the teacher a personalized course design space as an added value, as well as all the basic functionalities of an E-learning application.

In order to achieve this objective, we used different technologies mainly Node.js for the backend and React.js for the frontend.

Through this report, we show in detail the different steps taken to achieve this project.

# 1. Introduction

Le sujet traité dans ce rapport est le développement d'une application web e-learning offrant au professeur un espace de design de cours personnalisé.

Le e-learning tire son attrait du fait de pouvoir apprendre à son rythme, sur son ordinateur, des contenus pédagogiques sur des sujets variés. Organisée en sessions ou modules, avec tests d'évaluations, la formation peut-être totalement autogérée et suivie via un tableau de bord qui répertorie chacune des avancées du participant.

Le e-learning offre aussi aux formateurs un moyen efficace de dispenser des cours aux apprenants. L'apprentissage en ligne compte sur un certain nombre d'outils tels que les vidéos, les PDF, les podcasts, les tableaux blancs, etc. Les formateurs, qui peuvent aussi bien être des professeurs ou autres professionnels, peuvent utiliser tous ces outils dans le cadre de leurs plans de cours afin d'offrir des programmes d'études plus riches.

Une plateforme e-learning mené d'un éditeur de cours est ainsi un outil très puissant.

Le développement d'une telle plateforme est une chance pour appliquer les connaissances techniques acquises durant l'année scolaire précédente en matière de développement web ( HTML, CSS, JavaScript ), et de les améliorer en découvrant de nouveaux outils ( ReactJS, NodeJS, Express, MongoDB ).

C'est aussi une chance pour être en immersion dans le fonctionnement et la réalisation d'un projet en équipe avec des rôles spécifiques à chacun.

Ce rapport suivra le plan suivant : la conception de l'application, les choix techniques et leurs justifications, la structure de l'application, son implémentation et enfin une évaluation de cette expérience.

# 2. Analyse et conception

## 2-1. Introduction

La présente partie est consacrée à l'analyse et à la spécification des besoins du projet, indépendamment de toute technologie. L'étude consistera ici à déterminer, classer et prioriser les fonctionnalités du système. La spécification des besoins fonctionnels et non fonctionnels est la première phase qui a été entamée dans la réalisation de ce projet afin de concevoir un système de qualité qui répond à tous les besoins souhaités de notre plateforme E-learning tout en respectant bien les différentes contraintes.

## 2-2. Diagramme de bête à cornes

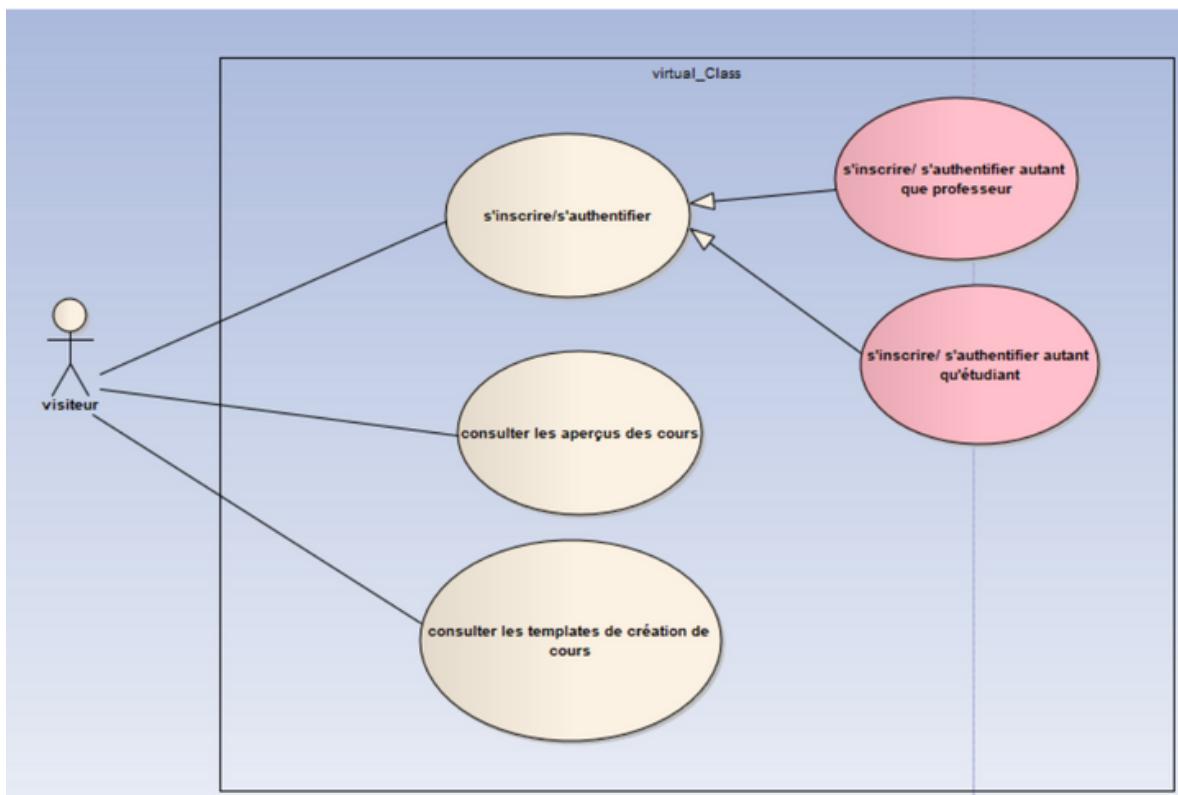


## 2-2. Besoins fonctionnels

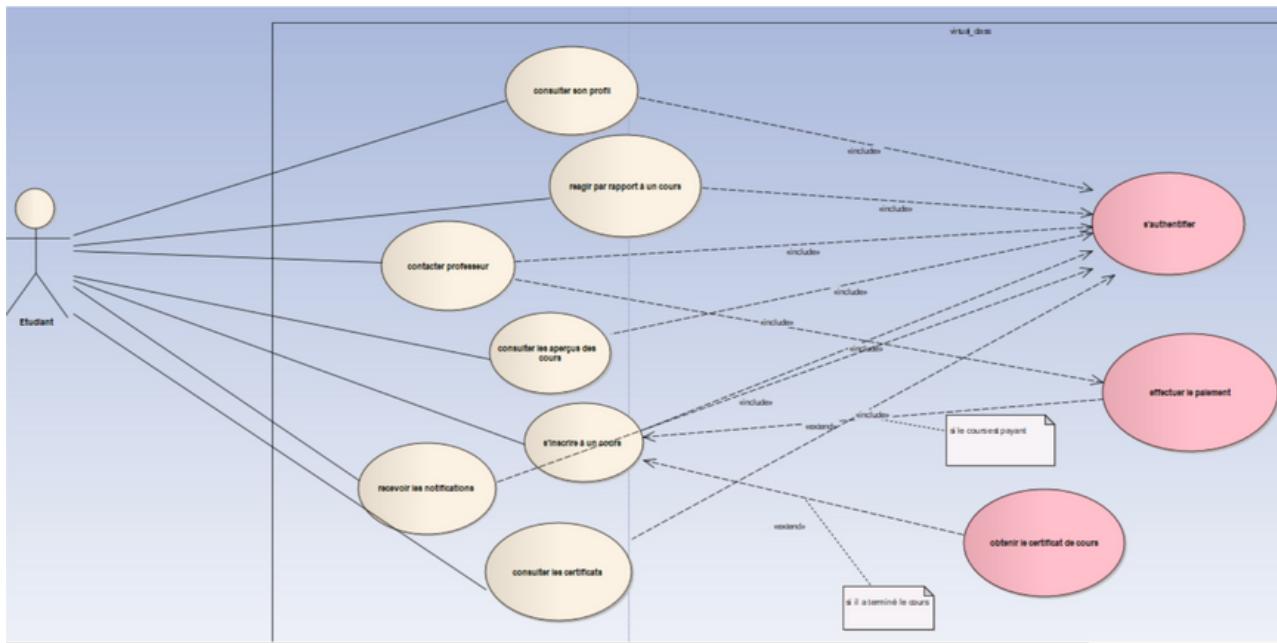
Pour spécifier les besoins fonctionnels et bien exposer ce que notre application offre en termes de services et de fonctionnalités, on a commencé par la réalisation des diagrammes de cas d'utilisation.

En guise de visibilité, on a créé un diagramme par acteur commençant par le visiteur non connecté qui peut:

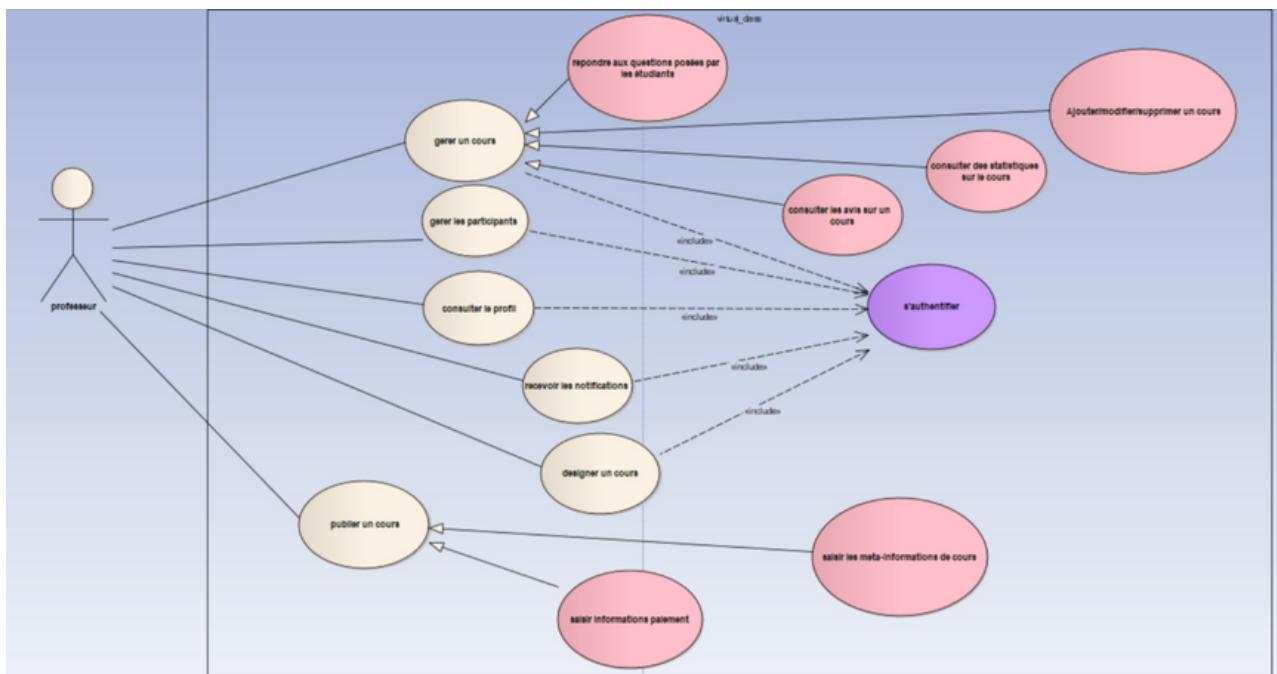
- S'inscrire ou s'identifier en tant que professeur ou en tant qu'étudiant;
- Consulter les aperçus des cours et des templates de création des cours.



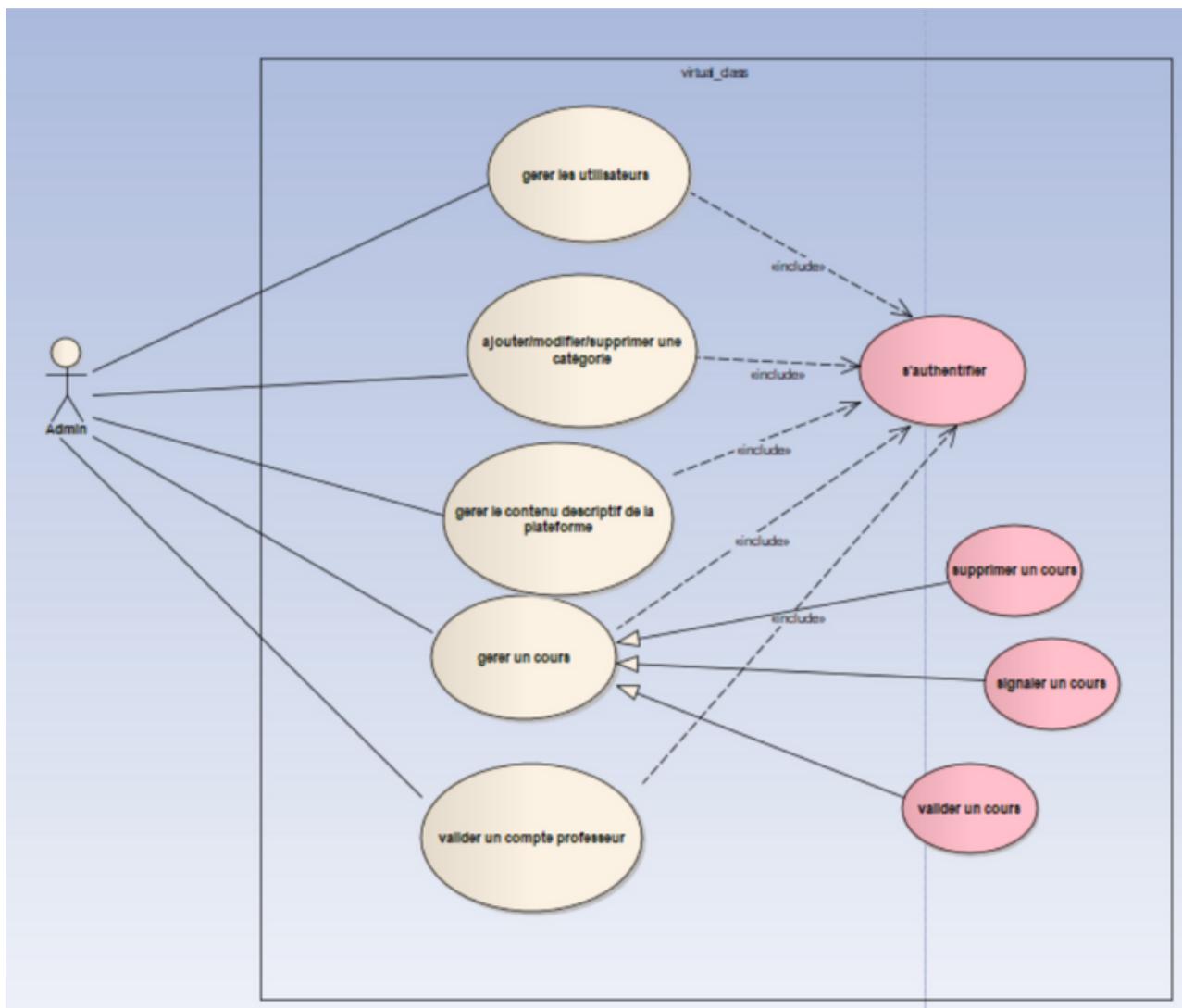
Si le visiteur s'inscrit et s'identifie comme étant un étudiant, il peut en plus de la fonctionnalité de la consultation des aperçus des cours offerte à un visiteur anonyme, s'inscrire à un cours en effectuant le paiement si le cours est payant, réagir par rapport aux cours en commentant et en notant les cours, contacter un professeur grâce au service de messagerie offert par notre plateforme, recevoir des notifications, consulter les certificats proposés et finalement consulter les informations de son profil :



Si au contraire, le visiteur s'inscrit comme étant un professeur, il peut créer un cours. C'est ce que représente en fait la valeur ajoutée de notre plateforme par rapport aux différentes plateformes E-learning déjà existantes; un professeur peut designer son cours soit "from scratch" ou en se servant d'une template grâce à un éditeur intégré à notre plateforme . Après avoir créé le cours, le professeur peut soit le sauvegarder comme "Draft" soit le publier un cours sur la plateforme. Le professeur peut gérer ses cours en consultant un tableau de bord qui offre les différentes statistiques, en ajoutant, supprimant et modifiant les cours, soit oublier la possibilité de consulter les participants aux différents cours. Un professeur peut également consulter son profil et recevoir des notifications :



Notre plateforme gère également la partie back office, en offrant à l'admin une interface pour gérer les utilisateurs, le contenu de la plateforme ( a propos de nous, les informations de contact..), les catégories des cours. Et c'est à l'admin aussi de valider le compte professeur pour assurer la crédibilité de la plateforme:



## 2-3. Besoins non fonctionnels

Étant une plateforme E-learning dédiée pour la création et l'inscription aux cours, Virtual class doit répondre à un certain nombre des besoins non fonctionnels indispensables :

- **La sécurité :** Une plateforme web doit avoir un certain niveau de sécurité parce que toute application qui n'est pas bien sécurisée est vulnérable et elle peut être exploitée par des attaquants pour effectuer des attaques sur notre plateforme ou sur nos visiteurs. Et plus particulièrement puisqu'on gère des paiements au niveau de la plateforme, on doit assurer un niveau avancé de confidentialité, d'intégrité et de disponibilité en implémentant différentes mesures et mécanismes.
- **Simplicité d'accès et d'utilisation :** On doit bien évidemment respecter les règles UI/UX design pour aboutir à une application efficace. Pour cela, la simplicité est une propriété très importante surtout que les professeurs qui vont créer des cours sur notre plateforme ne sont pas forcément familiers avec la technologie, d'où avoir un design intuitif est indispensable.
- **La maintenabilité :** Vu le manque de temps, on considère cette solution qu'un MVP qui doit être par la suite améliorée, ainsi la maintenabilité est très nécessaire et qui doit être assurée principalement par la visibilité du code source, la documentation des différentes étapes, la densité des commentaires et le respect des différentes bonnes pratiques.
- **Le référencement SEO :** Le marché des plateformes E-learning est un océan rouge plein des concurrents, d'où le référencement naturel SEO est un point critique pour la notoriété et le succès de Virtual class.
- **Accès offline :** Créer un cours pour un professeur prend beaucoup de temps, et pendant son travail, le professeur peut perdre la connectivité à tout moment, d'où garder son travail en manque ou en mauvaise connexion est très important.
- **Mobile friendly :** Actuellement, l'approche mobile first est largement adoptée, puisque la majorité des internautes utilisent le smartphone à la place de laptop, d'où avoir une plateforme responsive qui propose une bonne expérience utilisateur pour une personne qui se connecte en utilisant le mobile est prioritaire.

## 2-4. Conclusion

L'étude des besoins fonctionnels et non fonctionnels nous a permis de nous fixer des objectifs à atteindre afin d'assurer la bonne réalisation de la solution. On abordera puis détaillera dans le chapitre suivant les outils et technologies que nous avions choisi pour implémenter notre solution.

# 3. Mise en œuvre et réalisation

## 3-1. Introduction

Ce chapitre sera dédié à l'exposition des différents outils utilisés pour développer notre plateforme , tout en respectant au maximum les différentes exigences spécifiées auparavant , la justification des différents choix techniques effectués et la présentation de l'architecture technique sur laquelle on s'est basé pour implémenter notre solution.

## 3-2. Outils et technologies utilisés

Comme notre plateforme e-learning est implémentée sous forme d'une application web, les outils utilisés se basent sur les technologies du web, principalement les langages HTML et CSS pour le frontend et l'affichage du contenu dans le navigateur, et JavaScript pour la gestion de cet affichage mais aussi pour la gestion du backend, ou le travail du côté du serveur :

### **HTML**

HTML, ou HyperText Markup Language , désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. Il permet, entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu. Développé par le W3C (World Wide Web Consortium) et le WHATWG (Web Hypertext Application Technology Working Group), le format ou langage HTML est apparu dans les années 1990.L'HTML est ce qui permet à un créateur de sites Web de gérer la manière dont le contenu de ses pages Web va s'afficher sur un écran, via le navigateur. Il repose sur un système de balises permettant de titrer, sous-titrer, mettre en gras, etc., du texte et d'introduire des éléments interactifs comme des images, des liens, des vidéos... L'HTML est plus facilement compris des robots de crawl des moteurs de recherche que le langage JavaScript, aussi utilisé pour rendre les pages plus interactives.



## **CSS**

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Ce langage permet le maquillage des sites web créés par HTML, il permet de personnaliser la forme, les couleurs, la typographie ... Si HTML est indispensable pour créer un site web bien structuré, le CSS est ce qui permet de créer la partie UI des sites et les rend vifs.



## **Bootstrap**

Bootstrap est un framework CSS et est une compilation de plusieurs éléments et fonctions web-design personnalisables, le tout emballé dans un seul et même outil. Les développeurs qui utilisent Bootstrap pour la création de leur site web choisissent les éléments qu'ils veulent utiliser avec la certitude qu'ils ne seront pas incompatibles entre eux. En fait, c'est comme un puzzle. Sauf que dans ce puzzle, chaque pièce s'imbrique parfaitement dans les autres, quelle qu'elle soit. Les éléments personnalisables compilés dans Bootstrap sont une combinaison de HTML, CSS et JavaScript. Et grâce à la magie de l'open-source, Bootstrap s'améliore en permanence : de nouvelles fonctions absolument géniales ont été ajoutées comme l'aspect responsive (adaptation aux tailles d'écrans) ou la très large sélection de plugins jQuery .



## **JavaScript**

JavaScript est un langage de programmation destiné principalement au développement web. La particularité de ce langage est qu'il est très polyvalent, dans la mesure où on peut l'employer dans le frontend utilisant des bibliothèques comme jQuery combiné avec CSS afin de produire des éléments et des animations fluides.

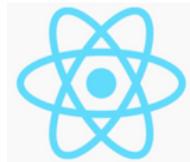
Toujours dans le frontend, JavaScript peut également servir d'outil de communication entre ce qui se passe dans le navigateur et le serveur qui devrait gérer le site web en question, et qui peut être aussi écrit en JavaScript ! Pour établir une telle communication, on utilisera le "Vanilla" JavaScript pour le frontend, "Vanilla" fait référence à l'utilisation du JavaScript pur sans avoir recours à des bibliothèques ou des frameworks ou en utilisant des boîtes à outils ou des frameworks front end tels que React JS, Angular JS et Vue JS.

Dans notre cas, on va utiliser la deuxième option et plus particulièrement React JS. On a également la possibilité d'utiliser Javascript pour les traitements côté serveur.



## React JS

ReactJS est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants réutilisables dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état. React offre un rendu côté client ce qui augmente les performances, mais les problèmes qui se posent avec React sont le chargement de la première page qui est relativement long et le référencement naturel SEO, puisque on n'a pas un code HTML qui est rendu en premier, alors on n'a pas le contenu sur lequel va se baser le Robot Crawler pour référencer les pages.



## Node.js/Express.js

**Node.js** est un environnement d'exécution JavaScript qui permet de créer rapidement des applications web. Node.js est principalement utilisé pour programmer du JavaScript du côté du serveur, avec une

Seule exigence est que le serveur n'effectue pas de lourdes opérations ( Node.js n'est pas très efficace pour le traitement de données par exemple ).

**Express.js** est un framework de Node.js qui a la particularité d'offrir encore plus de simplicité durant la création et la mise en place du fonctionnement du serveur. On y configure les routes que le client va

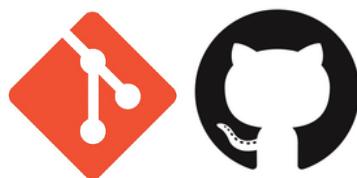
demander ( les demandes qui se feront par le JavaScript côté navigateur ), et ainsi on peut contrôler la manière de répondre.



## **Git/ GitHub**

Tout projet de développement subit des modifications et des rectifications et passe par des états ce qu'on appelle des versions, c'est là où l'utilisation de Git est indispensable : un système de contrôle de versions d'un projet donné. Il nous permet de marquer les différentes "versions" de notre projet, pour nous donner la possibilité de basculer facilement entre ces versions, tout ceci avec quelques commandes dans le terminal.

GitHub est une plateforme web basée sur Git, elle fournit une interface graphique contrairement à Git qui reste dans l'invite de commandes. Mais la principale utilisation de GitHub est qu'il facilite le travail de groupe, dans la mesure où ses membres peuvent librement modifier les fichiers du projet tout en restant à jour avec les modifications récentes. L'unique contrainte sera de bien coordonner entre les différents membres du groupe afin qu'un membre n'interfère pas avec le travail d'un autre.



## **MongoDB**

MongoDB est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++. Le serveur et les outils sont distribués sous licence SSPL, les pilotes sous licence Apache et la documentation sous licence Creative Commons 4. Il fait partie de la mouvance NoSQL. On a utilisé le MongoDB ATLAS qui représente une plateforme cloud permettant le déploiement des bases de données MongoDB.



## **Visual Studio Code**

Studio Code est un éditeur de code source développé par Microsoft pour Windows, Linux et MacOs, il inclut la prise en charge du débogage, du contrôle Git intégré et de GitHub, gratuit, supportant une dizaine de langages dont le langage Javascript, CSS et le langage de balisage HTML.



## **Enterprise Architect**

C'est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application. Ce logiciel est principalement payant mais il offre une période d'un mois pour l'essai.



## **Trello**

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. On a utilisé cet outil dans notre projet afin d'organiser les tâches.



## **Jira Software**

Jira est un système de suivi de bugs, de gestion des incidents et de gestion de projets développé par Atlassian et publié pour la première fois en 2002. Il propose des solutions à la fois à destination des développeurs et des intervenants non développeurs. On a utilisé Jira Software, le logiciel open source développé par Atlassian pour éditer le product backlog, le diviser en sprint backlog et diviser les user stories entre nous.



### 3-3. Justification des choix technologiques

Pour des raisons académiques en premier lieu, on a choisi de travailler avec le socle MERN qui englobe les technologies les plus utilisées à l'heure actuelle d'où travailler un projet académique dans ce sens est intéressant. Et bien sur il y'a d'autres raisons autre que celles académiques pour lesquelles on choisi de travailler avec ces technologies :



**ReactJS** : nous permet de créer des interfaces complexes à l'aide de composants simples et de les connecter aux données de notre serveur backend, les performances offertes par les applications créées en se basant sur ce framework, la richesse de React et sa maturité par rapport au Vue JS et sa simplicité par rapport à AngularJS sont parmi les causes principales pour lesquelles on a choisi React, sans oublier que c'est le framework JS le plus utilisé actuellement ce qui implique une réactivité intéressante au niveau des Forums React et une communauté active. Vu que notre application est une plateforme E-learning, le référencement SEO est une contrainte principale d'où on a pensé à utiliser Next JS qui représente un Framework React JS mais vu que nous ne sommes pas encore familiers avec React, c'est difficile d'entamer directement Next alors on a décidé de travailler premièrement avec React et migrer par la suite à Next.

**NodeJS**: Il fournit des applications multi plateformes qui fonctionnent facilement sur n'importe quel site web. Parmi les technologies Backend les plus puissantes utilisées actuellement, on trouve Spring Boot et Node JS. En les comparant, on trouve que Node est deux fois plus rapide que Spring, 45 % des fichiers moins, 33% du code moins d'où on comprend que Node est plus performant . Et pour notre contexte d'utilisation , E-learning, ces performances sont des points essentiels pour la réussite de la plateforme.

**MongoDB :** Elle peut supporter d'énormes volumes de données et de trafic. MongoDB offre une excellente expérience utilisateur pour les développeurs qui permet facilement de gérer des données structurées et celles non structurées. Pour raison d'évolutivité de notre plateforme, nous avons choisi de travailler dès le départ avec ce système de gestion de base de données qui supporte des données volumineuses et de différents types.

**Express.js :** Express.js dispose de modèles puissants pour le routage des URL et la gestion des demandes et réponses HTTP. Ce qui facilite énormément l'utilisation de NodeJS.

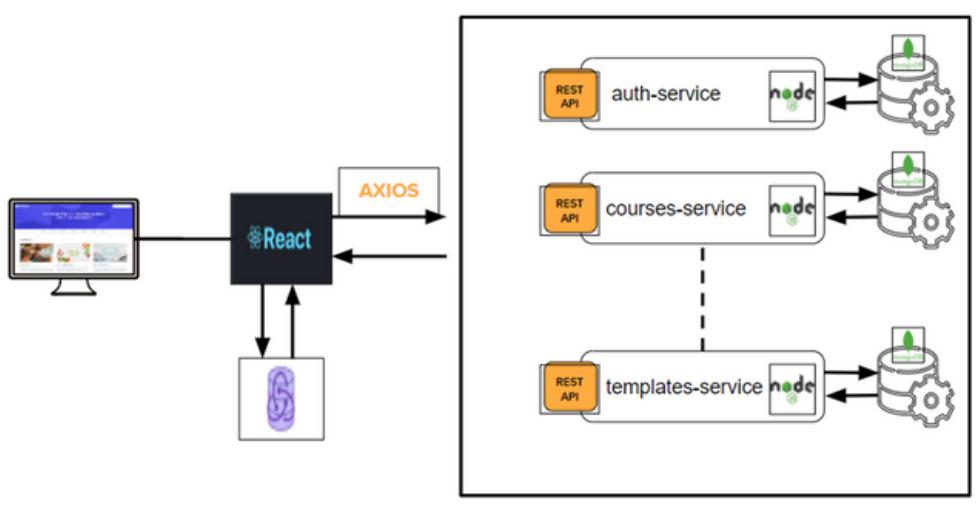
**Redux JS :** Redux est une bibliothèque open-source JavaScript de gestion d'état pour applications web. Elle est plus couramment utilisée avec des bibliothèques comme React ou Angular pour la construction d'interfaces utilisateur. Le fait de disposer d'un store unique pour gérer l'état global de l'application facilite grandement la visualisation et la compréhension du flux de données et de son incidence sur le comportement.

## 3-4. Structure de l'application

### Backend

Pour que notre application soit plus modulaire, uniforme et testable au fur et à mesure de la construction et pour qu'elle soit plus robuste, évolutive et disponible lors du déploiement dans l'environnement de production. On a opté de concevoir notre application Node.js comme ensemble de microservices indépendamment déployables:

```
> eureka-helper
<--> micro-services
    > auth-service
    > categories-service
    > courses-service
    > editor-service
    > messages-service
    > notifications-service
    > payments-service
    > templates-service
    .gitignore
    eureka-service-0.0.1-SNAPS...
    zuul-0.0.1-SNAPSHOT.jar
```



Dans une telle architecture, il est important d'intégrer un registre de découverte de services et de capacités de routage dynamique.

En fait nous aurons :

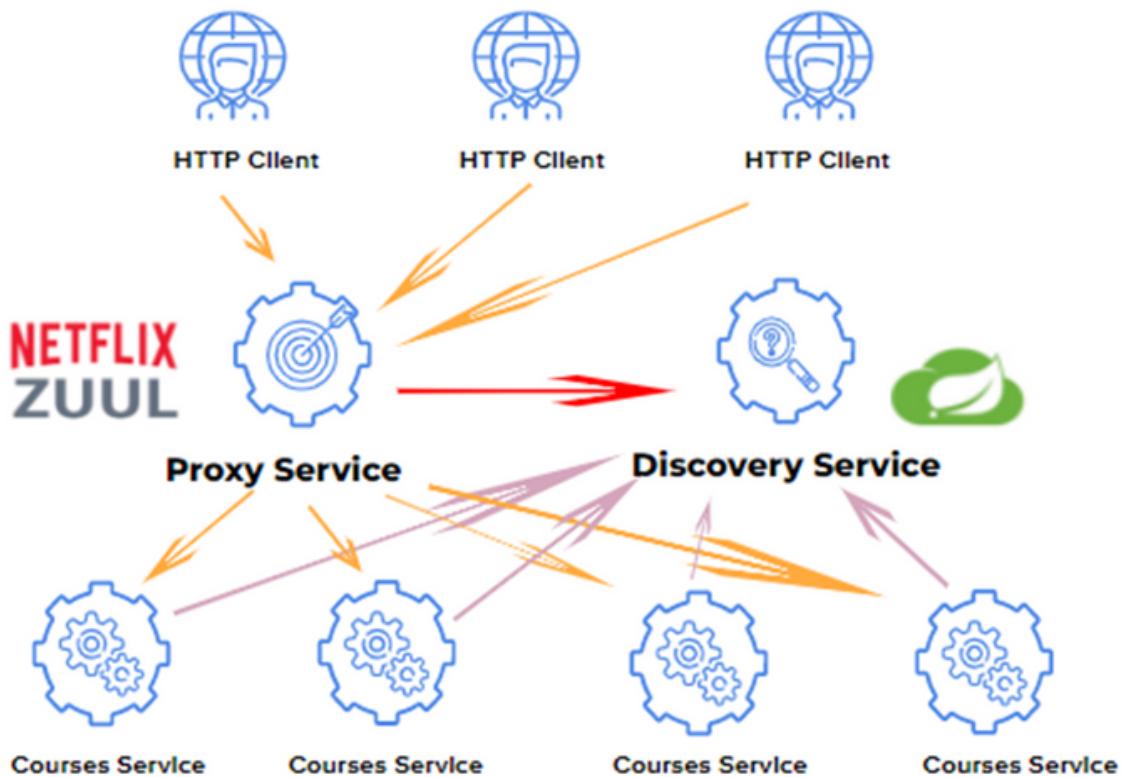
- Les microservices principaux comme auth-service, categories-service, courses-service ....chacun offrant une API REST pour gérer des fonctionnalités dédiées.
- Proxy Service : Passerelle se chargeant du routage d'une requête vers l'une des instances d'un service, de manière à gérer automatiquement la distribution de charge. En effet, L'architecture microservices, en fournissant un ensemble de services indépendants et faiblement couplés, se trouve confrontée au challenge de fournir une interface unifiée pour les consommateurs, de manière à ce qu'ils ne voient pas la décomposition à faible granularité de vos services. C'est pour cela que l'utilisation d'un service proxy, responsable du routage des requêtes et de la répartition de charge, est important.

```

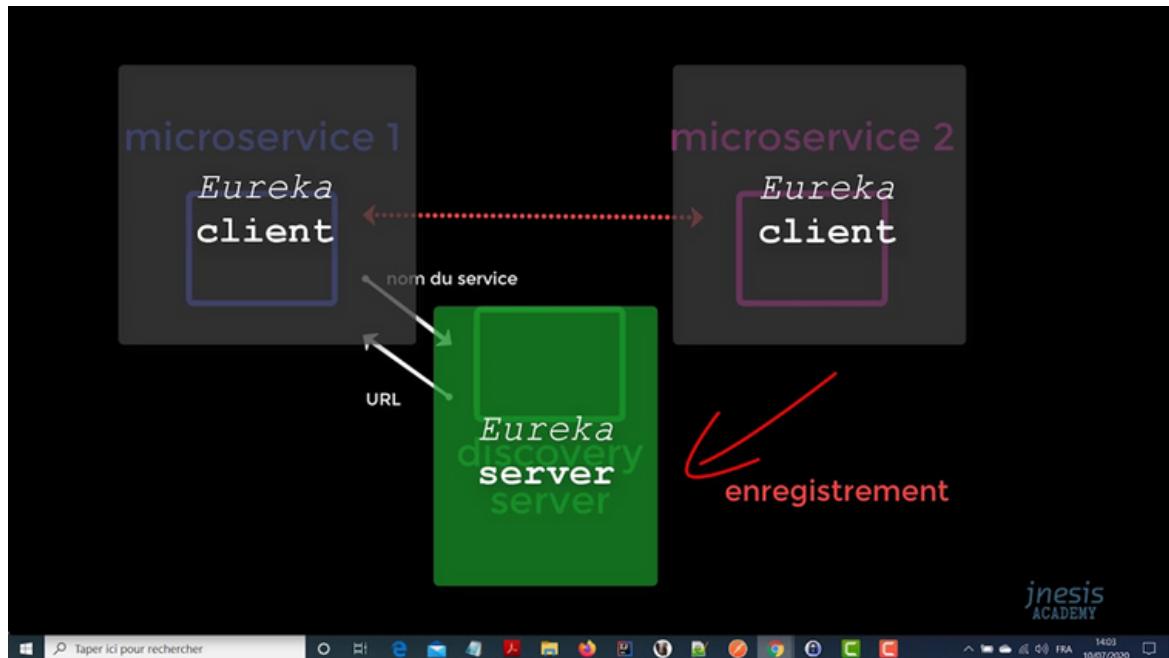
courses-service
├── config
│   ├── .env
│   └── db-config.js
├── node_modules
└── src
    ├── controllers
    ├── middlewares
    ├── models
    ├── routes
    ├── index.js
    ├── package-lock.json
    └── package.json

```

- **Discovery Service:** Service permettant l'enregistrement des instances de services en vue d'être découvertes et exploitables par d'autres services. En effet, pour éviter un couplage fort entre microservices, on utilise un service de découverte qui donne aux services de notre application la possibilité de trouver d'autres services sans savoir où ils sont hébergés ou l'URL complète requise pour les atteindre. Le service de découverte fournira dynamiquement les informations nécessaires, ce qui permet d'assurer l'élasticité et la dynamicité propres à notre architecture microservices.



- Pour ce faire, dans notre cas on s'est basé sur un projet open source offrant un serveur de découverte Netflix Eureka implémenté par Spring Boot.
- On s'est servi alors du fichier .jar pour Eureka disponible dans le cadre du dépôt Spring Guides sur GitHub pour intégrer Eureka Server dans notre application.



D'autre côté, on a implémenté un mécanisme d'enregistrement pour l'appeler dans chaque micro service comme étant Eureka Client.

```
const start = async () => {
  try {
    await server.listen(port, () => { console.log(`Server started on ${port}`) });
    await eureka.registerWithEureka('auth-service', port);
    connectDB();
  }
```

Ainsi, une fois services sont enregistrés auprès d'Eureka, ils peuvent être localisés grâce aux capacités de routage dynamique de Zuul, afin qu'on puisse accéder à tous les microservices en utilisant le même URL pour assurer l'homogénéité de l'application, ce qui est bien montré au niveau des captures des tests qu'on a effectué en utilisant Postman:

```
PS C:\Users\acer\Dev Web\virtual-class-microservices-mern> java -jar zuul-0.0.1-SNAPSHOT.jar
--server.port=9090 --eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka/
```

**http://localhost:8002/api/courses**

GET <http://localhost:8002/api/courses> Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description	...	

Body Cookies Headers (8) Test Results 200 OK 97 ms 3.94 KB Save Response

Pretty Raw Preview Visualize JSON

```

1
2   "courses": [
3     {
4       "followers": [],
5       "id": "623255685f6f336007h540d2"

```

**http://localhost:9090/courses-service/api/courses**

GET <http://localhost:9090/courses-service/api/courses> Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description	...	

Body Cookies Headers (8) Test Results 200 OK 69 ms 3.97 KB Save Response

Pretty Raw Preview Visualize JSON

```

1
2   "courses": [
3     {
4       "followers": [],
5       "id": "623255685f6f336007h540d2"

```

POST http://lo X GET http://lo ● GET http://lo ● GET http://lo ● POST http://lo ● + ... No Environment

**http://localhost:9090/auth-service/api/login**

POST <http://localhost:9090/auth-service/api/login> Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description	...	

Body Cookies Headers (9) Test Results 200 OK 160 ms 884 B Save Response

Pretty Raw Preview Visualize JSON

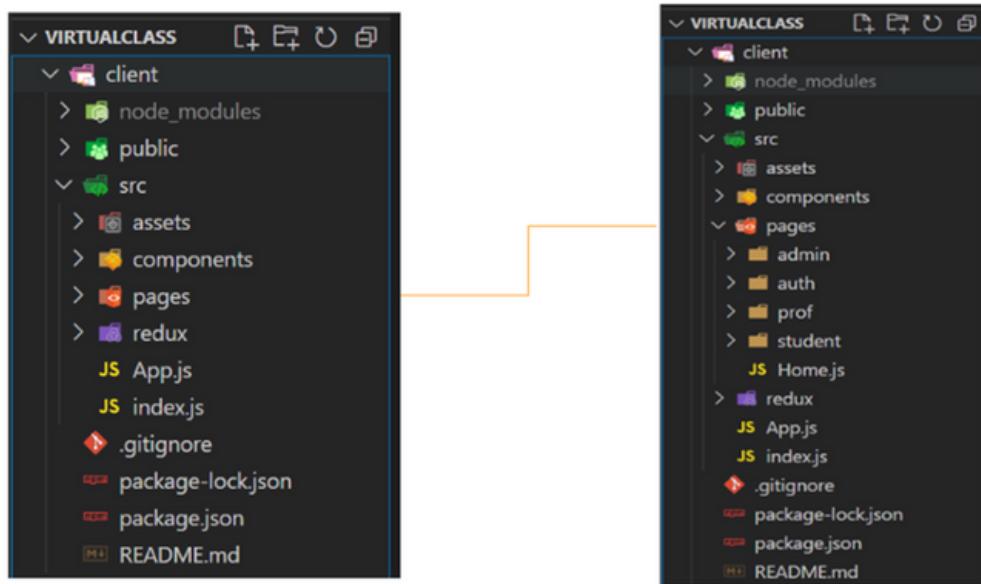
```

1
2   "message": "User logged in",
3   "user": {
4     "_id": "621e592316735a536531c941",
5     "first_name": "test",
6     "last_name": "test",
7     "email": "teststudent0@test.com",
8     "password": "$2b$10$fiFFmc98fk9bPMPGBH5Ke00th/o.mCepc320WxZlWnnDDawuakhXQ0",
9     "salt": "vA1^..r"

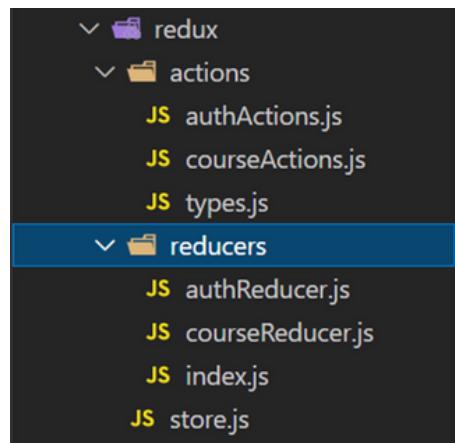
```

## Frontend

Pour le côté frontend, on a personnalisé la structure d'un projet React en ajoutant créant un ensemble des dossiers pour l'organisation des fichiers comme c'est montré dans la capture ci-dessous :



L'utilisation de Redux : Comme c'est déjà expliqué au niveau de la partie justification des choix technologiques, on a exploité Redux pour gérer les états et les données de la plateforme d'une manière efficace, pour cela on à rassembler l'ensemble des actions au niveau d'un dossier sous le dossier de Redux:



## Conclusion

Nous avons pu analyser dans ce chapitre en détail l'architecture de ce projet (backend et frontend) qui n'est pas, malgré l'utilisation de plusieurs outils et l'intégration de différents concepts, si complexe comme elle le paraît. Dans le prochain chapitre on parlera de la création de la base de données.

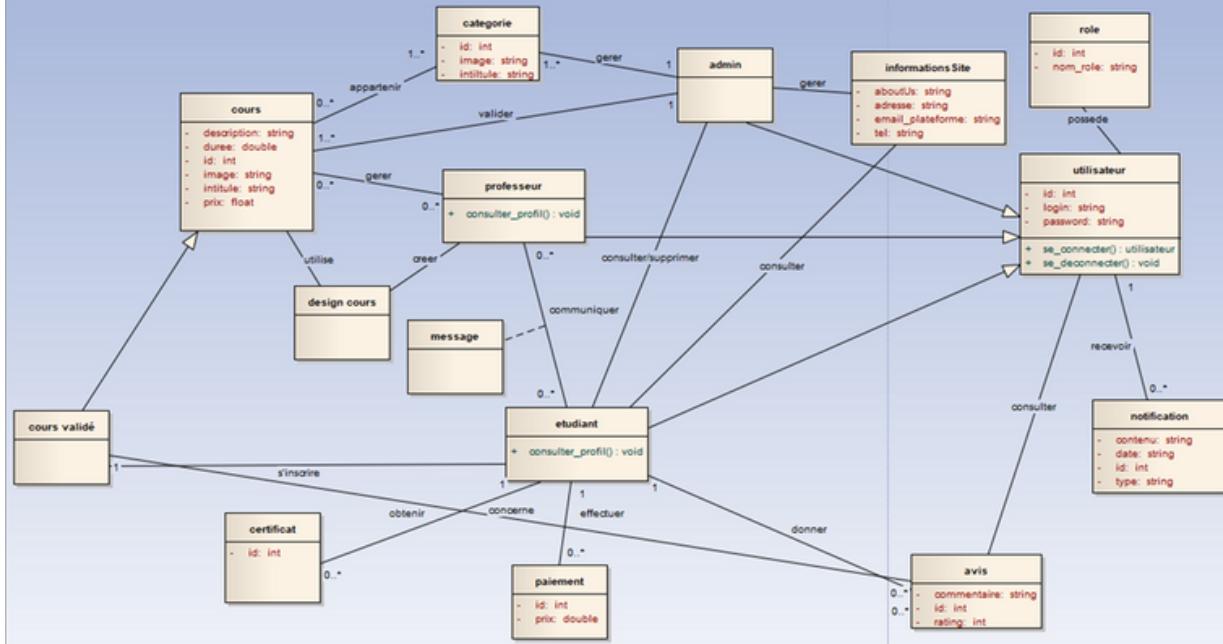
# 4. Cration de la base de donnes

## 4-1. Introduction

Comme c'est déjà mentionné au niveau des justifications des choix technologiques, nous avons choisi de travailler avec MongoDB pour la création de notre base de données, et plus particulièrement nous avons décidé de travailler avec MongoDB ATLAS comme étant hébergeur des bases de données en cloud.

## 4-2. Diagramme de classe:

Afin de décrire clairement la structure de notre application, c'était nécessaire de créer le diagramme de classe et l'enrichir le plus possible par les attributs et les opérations afin d'avoir une visibilité sur la structure de la plateforme:



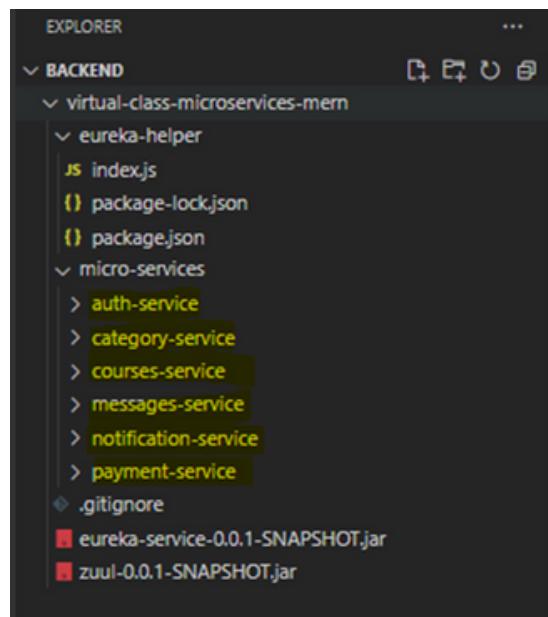
## 4-3. Cration des modles

Premièrement, on a configur la connexion au cluster MongoDB hberg au niveau de MongoDB ATLAS en prcisant quune erreur doit tre affiche si la connexion nest bien ´tablie:



```
JS db-config.js X
virtual-class-microservices-mern > micro-services > auth-service > config > JS db-config.js > ...
1
2 const mongoose = require('mongoose');
3 const URI = process.env.URI_DB;
4 require('dotenv').config({ path: '../config/.env' });
5
6
7 const connectDB = async () => {
8     try {
9         await mongoose
10            .connect(
11                URI,
12                {
13                    useNewUrlParser: true,
14                    useUnifiedTopology: true
15                }
16            )
17            .then(() => console.log('MongoDb connected...'))
18        } catch (error) {
19            console.log(error)
20        }
21    }
22
23 module.exports = { connectDB }
```

Par la suite on implmente les diffentes modles en prcisant le schma de chacun :



Par exemple, on a les schéma de user est défini comme est montré dans la capture ci-dessous, les autres modèles sont établies en suivant le même logique:

```
const mongoose = require('mongoose');
const { isEmail } = require('validator');
const bcrypt = require('bcrypt');

const userSchema = new mongoose.Schema(
{
  first_name: {
    type: String,
    lowercase: true,
  },
  last_name: {
    type: String,
    lowercase: true,
  },
  email: {
    type: String,
    required: true,
    validate: [isEmail],
    lowercase: true,
    unique: true,
    trim: true,
  },
  password: {
    type: String,
    required: true,
    max: 1024,
    minlength: 6
  },
  role: {
    type: String,
    required: true
  }
},
{ timestamps: { createdAt: 'created_at', updatedAt: 'updated_at' } }); //automatically add while insert or update the object
```

## 4-4. Conclusion

Avoir une base de données bien faite est indispensable pour développer correctement l'application.C'est pour cela, on a bien investi du temps à bien élaborer les modèles. Dans le chapitre suivant, on abordera la méthodologie de travail.

# 5. Mise en œuvre de la solution

## 5-1. Introduction

Nous nous intéresserons dans ce chapitre à la méthodologie de travail adoptée. On a opté pour une méthodologie de travail agile qui consiste à développer la solution d'une manière itérative et incrémentale:



Et parmi les méthodologies agile, on a choisi de travailler avec SCRUM qui représente une approche empirique et participative permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière créative des produits de la plus grande valeur possible « Guide SCRUM ». Scrum à trois piliers et 5 valeurs:



## 5-2. Crédit d'un product Backlog:

Au niveau de Jira Software, on a créé notre product backlog qui contient toutes les user stories. Puis on les divise à des sprint backlogs, et par la suite à des tâches qu'on a organisées au niveau du Trello.

Projets / virtual-class-team

## Backlog

Story	Statut
VCT-1 En tant que visiteur, je veux consulter la liste des cours pour avoir une idée générale sur les cours proposés.	À FAIRE
VCT-2 En tant que visiteur, je veux consulter un aperçu de cours pour décider de le choisir ou pas.	À FAIRE
VCT-3 En tant que visiteur, je veux filtrer les cours par prix, par durée, par les plus récents pour me faciliter le choix du cours à suivre	À FAIRE
VCT-4 En tant que visiteur, je veux choisir la catégorie des cours à suivre pour me focaliser sur un domaine bien précis.	À FAIRE
VCT-7 En tant qu'étudiant, je veux acheter un cours payant pour le commencer.	À FAIRE
VCT-8 En tant qu'étudiant, je veux donner mon avis sur un cours, le commenter et le liker pour pouvoir y réagir.	À FAIRE
VCT-9 En tant qu'étudiant, je veux poser des questions sur un cours que j'ai déjà acheté pour bien le comprendre.	À FAIRE
VCT-10 En tant qu'étudiant je veux accéder à mon profil pour le modifier.	À FAIRE
VCT-11 En tant qu'étudiant, je veux être notifié des nouveautés, des réponses des professeurs...pour être à jour.	À FAIRE
VCT-13 En tant que professeur, je veux consulter les avis des étudiants sur un cours et le nombre d'étudiants qui l'ont acheté pour booster ma productivité.	À FAIRE
VCT-14 En tant que professeur, je veux être notifié que mon cours a été validé.	À FAIRE
VCT-15 En tant que professeur, je veux être notifié qu'un étudiant a posé une question pour pouvoir le répondre au plus tard possible.	À FAIRE
VCT-16 En tant que professeur, je veux ajouter des collaborateurs pour gérer un cours.	À FAIRE
VCT-17 En tant que professeur, je veux consulter mon profil pour le modifier.	À FAIRE
VCT-19 En tant qu'administrateur, je veux consulter, ajouter, supprimer et modifier les catégories des cours.	À FAIRE
VCT-21 En tant qu'administrateur, je veux consulter la liste des utilisateurs.	À FAIRE
VCT-22 En tant qu'administrateur, je veux supprimer un utilisateur qui porte un risque à l'application	À FAIRE
VCT-23 En tant qu'administrateur, je veux valider les cours pour que leurs créateurs peuvent les publier sur l'application.	À FAIRE
VCT-24 En tant que professeur, je veux créer une template à mon cours pour le personnaliser	À FAIRE
VCT-25 En tant que professeur, je veux fixer les prix de mon cours.	À FAIRE
VCT-26 En tant que professeur, je veux gérer les participants à mon cours.	À FAIRE
VCT-27 En tant qu'administrateur, je veux valider l'identité des professeurs voulant s'inscrire .	À FAIRE

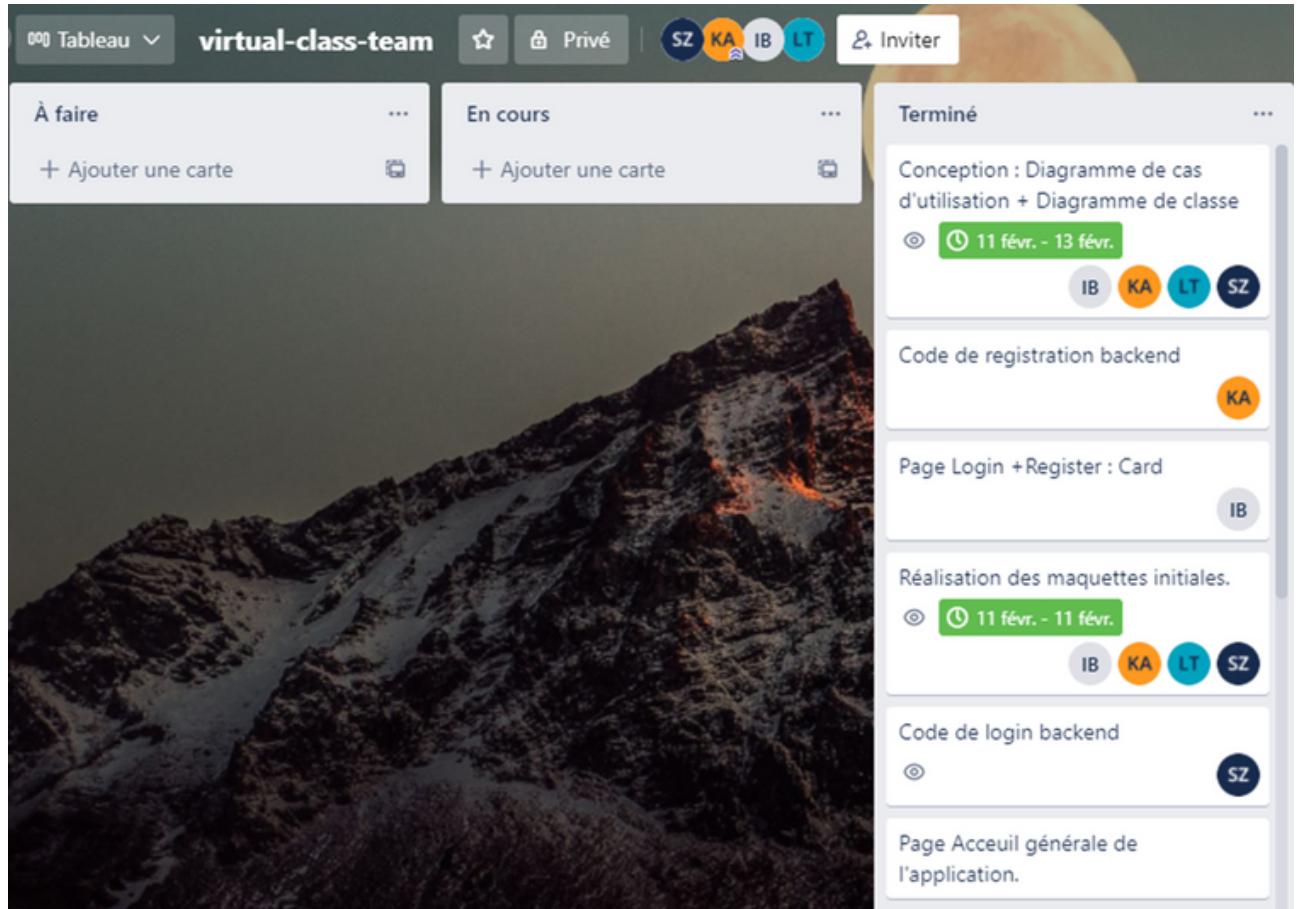
### 5-3. Crédit des sprints Backlog:

Après avoir établi toutes les user stories, on les divise à des sprint backlog, qui représente la liste des user stories qui doivent être réalisées pendant le sprint:

Story	Statut
VCT-5 En tant que visiteur, je veux m'inscrire pour suivre des cours.	TERMINÉ(E)
VCT-6 En tant qu'étudiant, je veux m'authentifier pour m'inscrire à des cours.	TERMINÉ(E)
VCT-10 En tant que professeur, je veux m'authentifier pour gérer (consulter,ajouter, modifier, et supprimer) mes cours.	TERMINÉ(E)
VCT-12 En tant qu'administrateur, je veux m'authentifier pour gérer le contenu descriptif de l'application.	TERMINÉ(E)

### 5-4. Division des user stories du sprint à des tâches:

Parfois les user-stories ne sont pas de même importance alors on est obligé de les diviser à des tâches et gérer ces derniers en utilisant Trello:



## 5-5. Exploitation de système de gestion des versions GIT/GitHUb :

L'implémentation de l'architecture technique nécessitait une très bonne coordination entre les membres du groupes, de telle façon à ce qu'un membre n'interfère pas avec le travail d'un autre tant que ce dernier

N'aie pas terminé. Git et GitHub nous ont donné la possibilité de bien gérer ce travail de groupe, par le biais des fetch et push qui permettent de mettre à jour l'environnement de développement de chaque collaborateur, et dans le cas où l'un d'eux désire valider ces changements, il n'aura qu'à demander un pull request, cette étape consiste à faire une comparaison entre ce qui a été présent avant et ce qui est sujet à l'ajout, afin de permettre une intégration souple des travaux entre eux, sans pour autant qu'il y ait des conflits entre eux. Dans cet exemple ci-dessous, on peut voir explicitement les modifications pour un fichier donné. Les lignes rouges indiquent les suppressions tandis que les lignes vertes montrent les ajouts. L'intégration se fait d'une manière aisée et assurée par Git, en maintenant toujours le principe de versions qui nous permet de revenir en arrière ou de rétablir des changements à tout moment. On a créé deux repository: un pour le frontend et l'autre pour le backend:



The image displays two screenshots of GitHub project pages side-by-side.

**Top Screenshot (salima-2000/virtualClass-frontend):**

- Branches:** master (4 branches, 0 tags)
- Commits:** ikrame007 Update (4 commits last month)
- README.md:** Getting Started with Create React App
- Available Scripts:** npm start
- Contributors:** salima-2000, ikrame007 (Ikrame Benjeddi), AMALK319

**Bottom Screenshot (AMALK319/virtual-class-microservices-mern):**

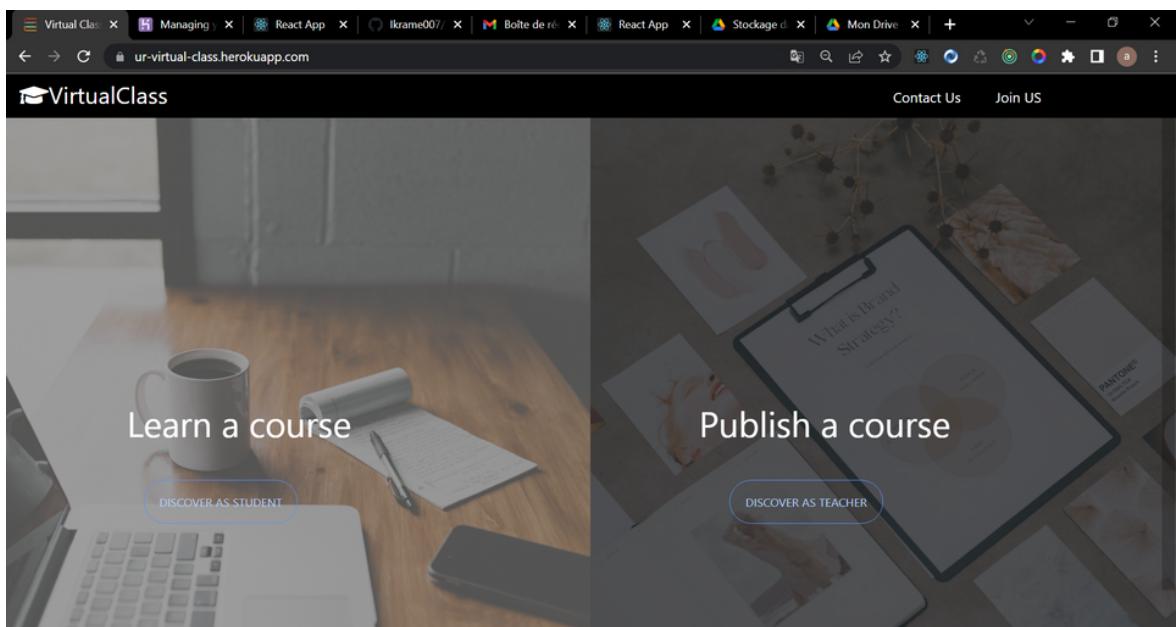
- Branches:** main (3 branches, 0 tags)
- Commits:** AMALK319 last commit before first delivery (19 commits, cb8f571, 5 minutes ago)
- Contributors:** No description, website, or topics provided.

## 5-6. Conclusion :

La méthodologie de travail est un point clé pour la réussite de tout projet. Dans ce cadre, adopter une méthodologie agile est très avantageux, et il permet de produire rapidement et avec efficacité d'une manière collaborative. Pour ce même objectif de collaboration, travailler avec le système de gestion de versions est indispensable pour bien gérer les changements et les rectifications tout en long le cycle de développement de notre plateforme.

# 6. Implémentation

## 6-1. Page d'accueil



La première page représentant la page d'accueil de notre plateforme dans laquelle l'utilisateur est face à deux choix :

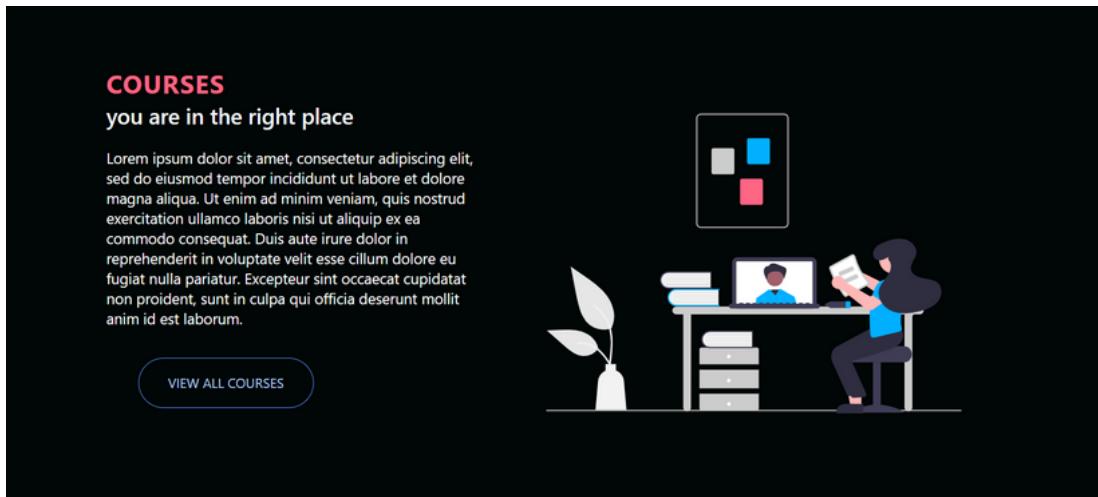
- découvrir en tant qu'étudiant
- découvrir en tant que professeur

Il peut aussi consulter nos informations générales en cliquant sur "About", comme il peut nous contacter avec "Join us" .

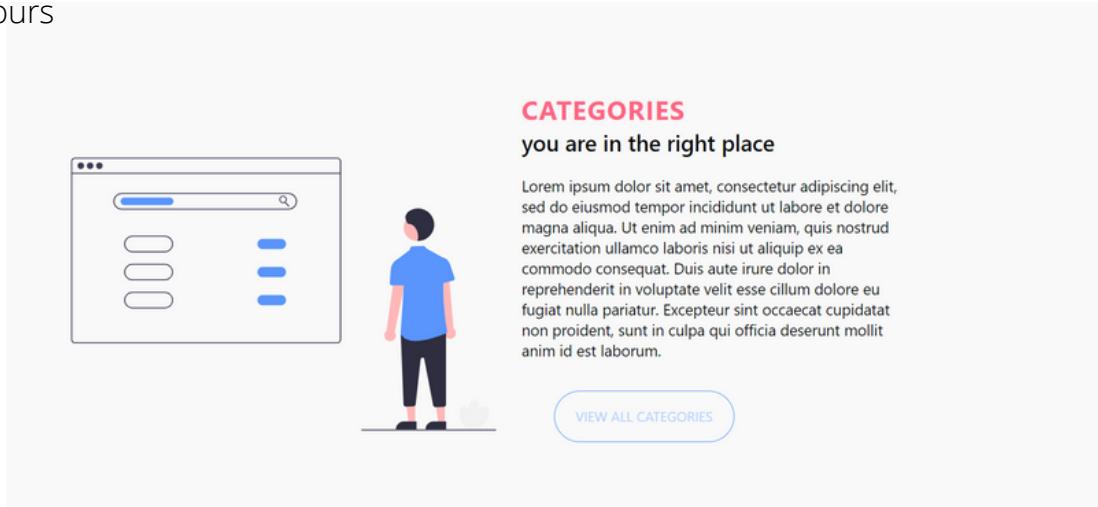
## 6-2. Espace étudiant :

La page de découverte pour un étudiant illustre une description des trois fonctionnalités de base offerte dans l'espace étudiant :

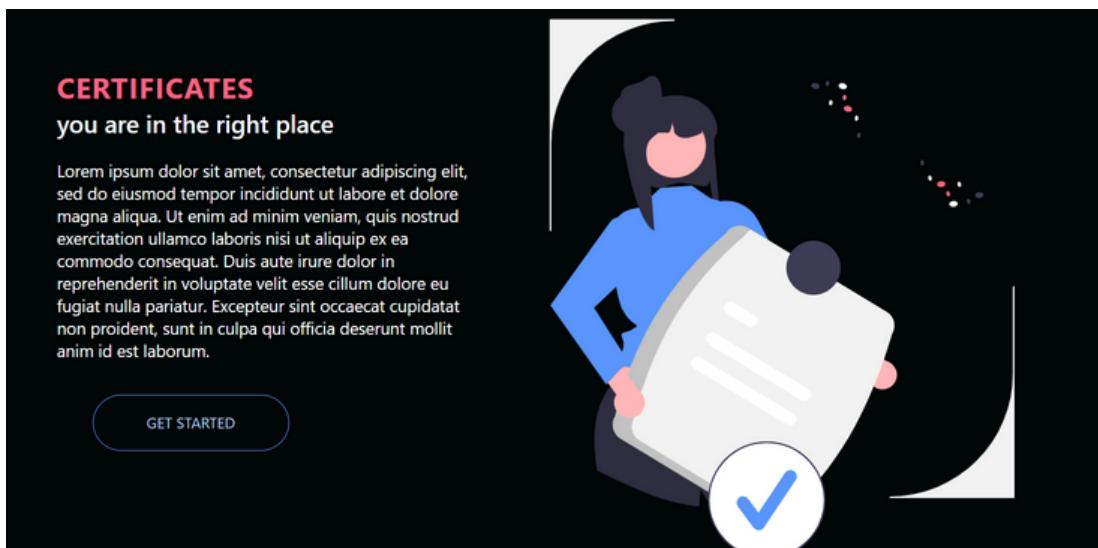
- **Les cours :** l'étudiant peut accéder au différents cours partagés par les professeurs



- **Les catégories :** il a aussi la possibilité de consulter les multiples catégories des cours



- **Les certificats:** finalement il peut avoir des certificats pour n'importe quel cours .



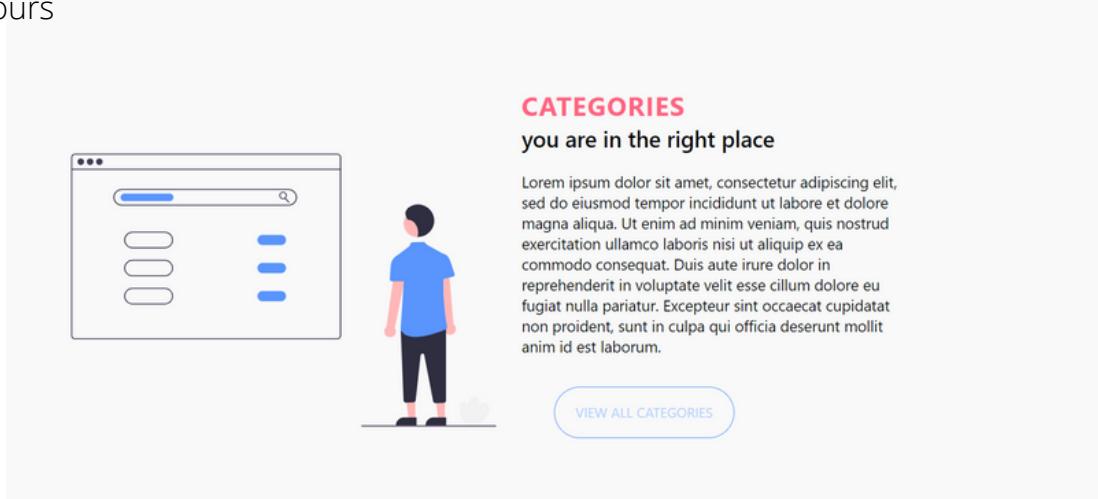
## 6-2. Espace étudiant :

La page de découverte pour un étudiant illustre une description des trois fonctionnalités de base offerte dans l'espace étudiant :

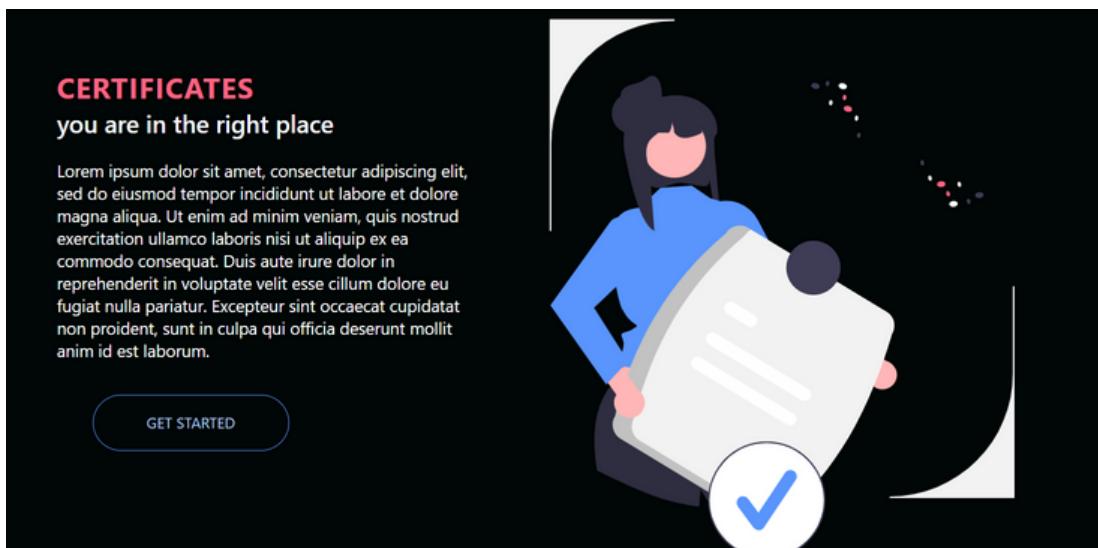
- **Les cours :** l'étudiant peut accéder au différents cours partagés par les professeurs



- **Les catégories :** il a aussi la possibilité de consulter les multiples catégories des cours



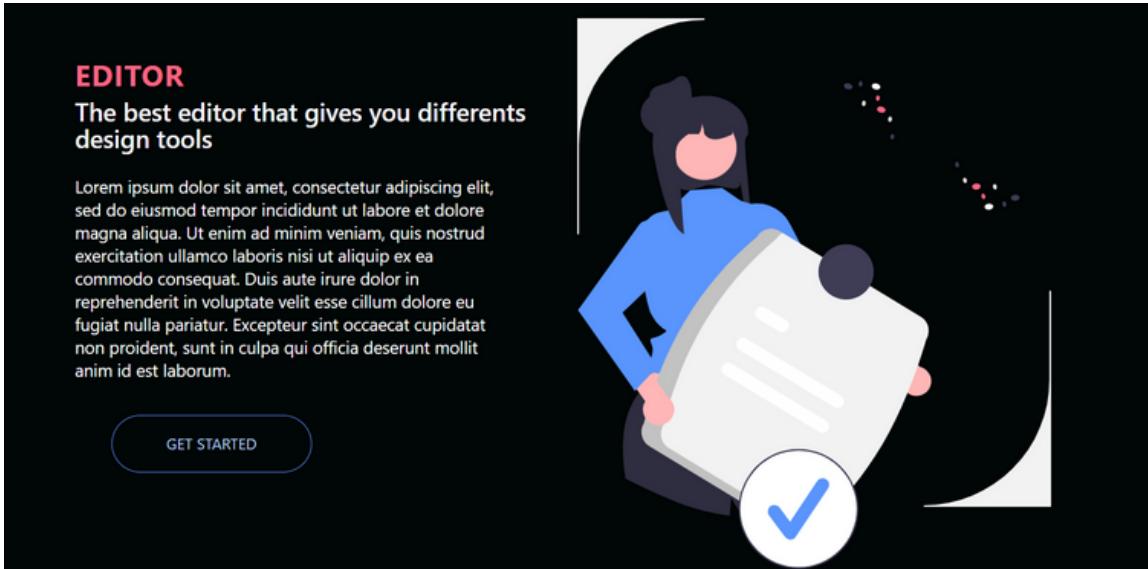
- **Les certificats:** finalement il peut avoir des certificats pour n'importe quel cours .



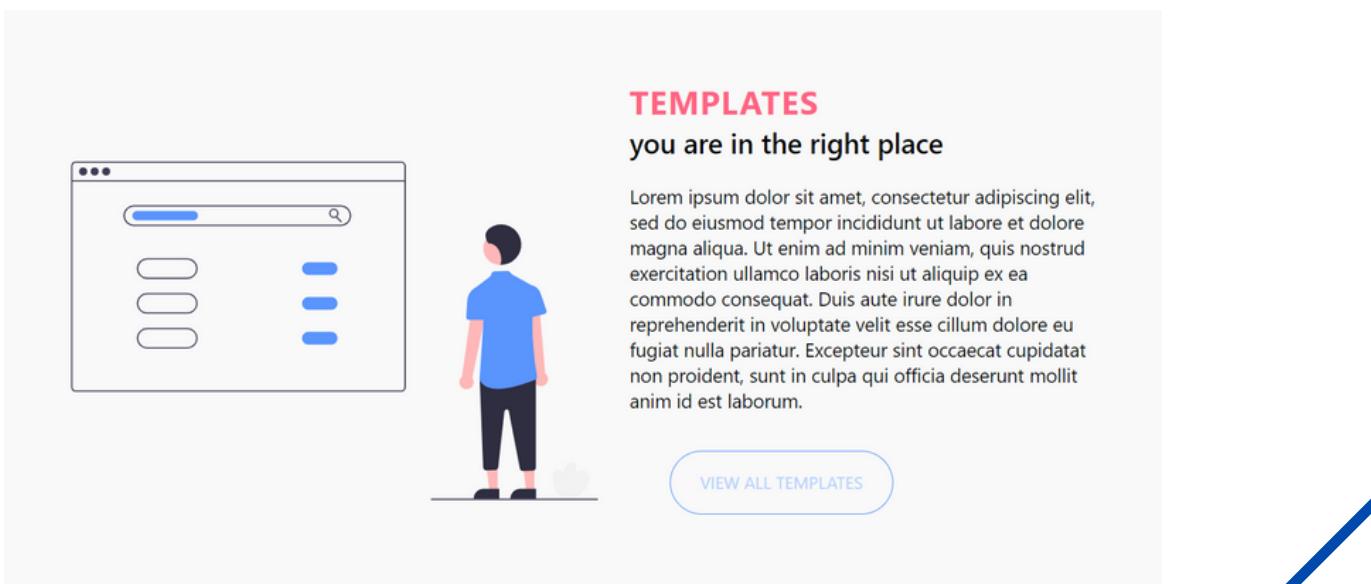
## 6-3. Espace professeur :

La page de découverte pour un professeur montre elle aussi une description des deux fonctionnalités de base offerte dans l'espace professeur :

- **L'éditeur:** il s'agit d'un espace propre au professeur où il peut personnaliser son cours avec des outils et des palettes.

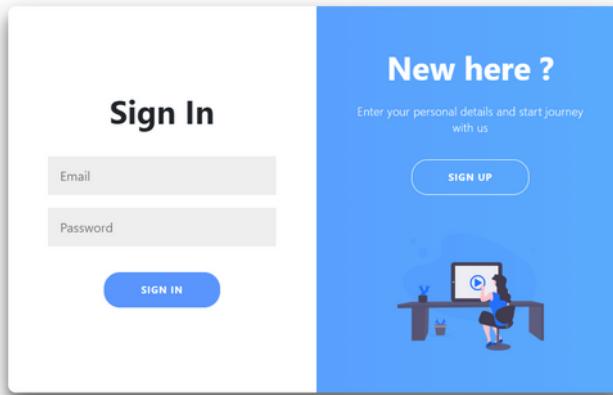


- **Les templates :** le professeur peut consulter des designs de cours déjà faites afin de les utiliser ou bien juste pour s'inspirer.

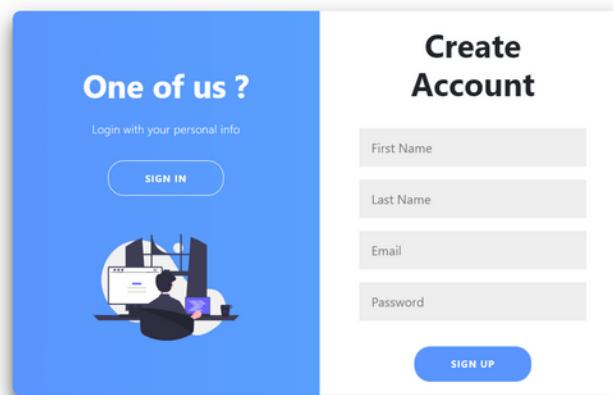


## 6-4. Authentification:

En cliquant sur Join us l'utilisateur sera rediriger vers une page de connexion dans laquelle il est censé saisir son E-mail ainsi que son Mot de passe .



Et dans le cas ou il ne possède pas un compte, il pourra le créer facilement en remplissant ses propres informations et en cliquant sur le bouton Sign up.



## 6-5. Consultation des cours

La page des cours permet de lister les meilleurs cours ,ainsi que ceux gratuits . L'utilisateur peut également effectuer des filtrage selon plusieurs volets :

- Difficulté
- Prix
- Temps

En plus de la fonctionnalités de recherche par un mot clé .

The screenshot shows a course detail page for 'React.js | Development Web'. At the top, there's a navigation bar with 'VirtualClass' logo, 'Courses', 'Categories', 'Login', 'Home', 'About', and 'Join Us'. Below the navigation is a large image of a person working on a computer with a Node.js logo on the screen. A green hexagonal overlay contains the text: 'Reusability is key in reducing bugs and coding quickly.' To the right of the image, the course title 'React.js | Development Web' is displayed with a 'New' badge. Below the title is a short description: 'Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un imprimeur anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.' Below the description is the author 'By Jean Guillaume' and a 5-star rating '(202 notes)'. The price '25.00\$' is listed, followed by a 'Check Course' button.

Difficulty      Price      Time      Search...

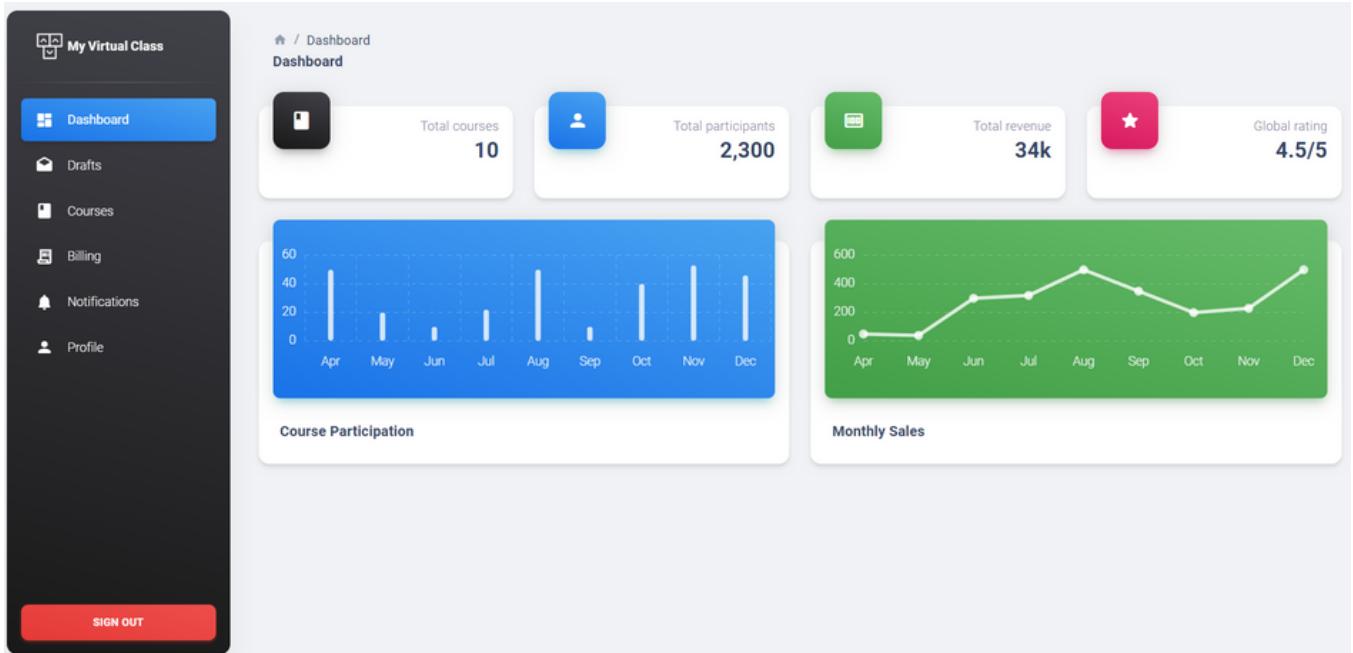
The screenshot shows a section titled 'Top Courses Of The Week' featuring three identical course cards for 'ReactJS TRAINING'. Each card includes the course name, 'Development Web' category, author 'By Jean Guillaume', a 5-star rating, a price of '25.00\$', and a 'Check Course' button.

### Get Started with Free Courses



## 6-6. Dashboard professeur :

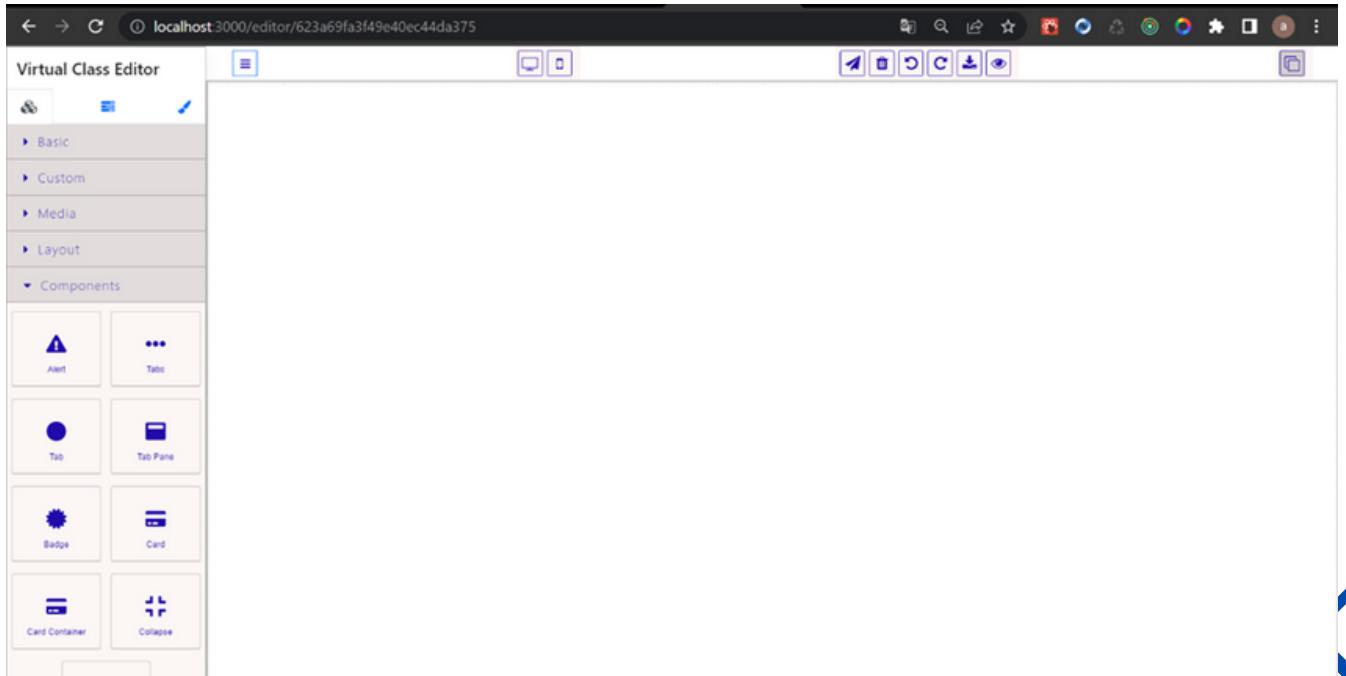
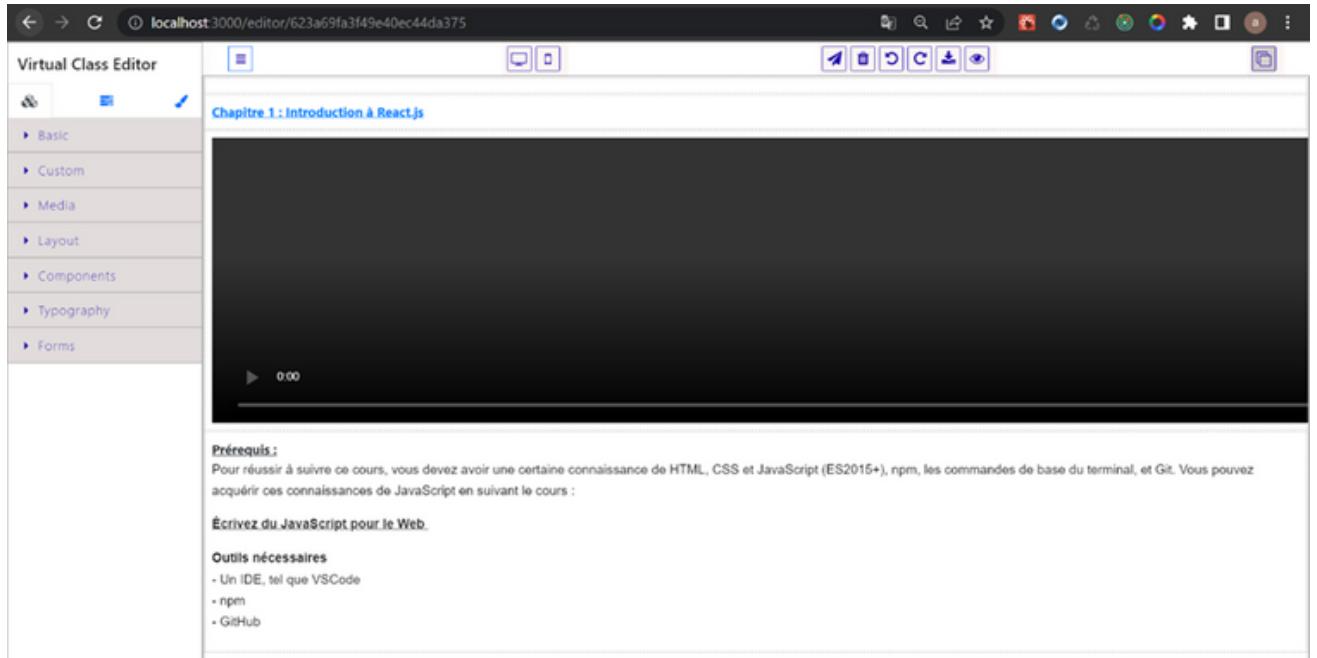
Après connexion, le professeur peut consulter son dashboard qui comprend un overview, ses brouillons, ses cours, les informations de son compte bancaire et son profil.



## 6-7. Editeur :

L'éditeur comporte plusieurs palettes et dans chacune différents outils permettant ainsi au professeur de personnaliser son cours et de le rendre plus claire et plus enrichissant par l'ajout des images, vidéos..

Et parmi les fonctionnalités principales qu'il peut choisir lesquelles des parties de son cours il préfère rendre visible aux étudiants et lesquelles rendre cachées.



## 6-8. Lien du "GitHub public repository" contenant le code source de l'application

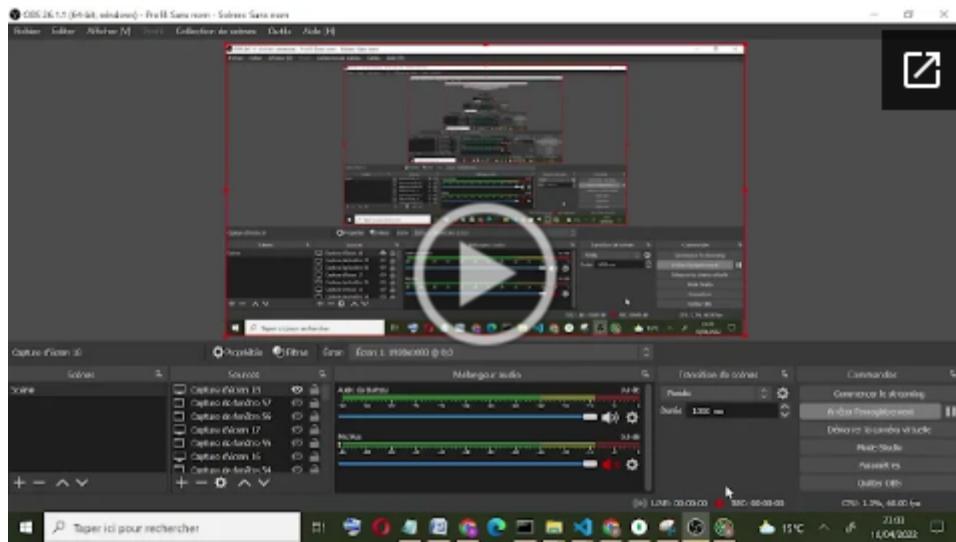
### **Branche main**

<https://github.com/lkrame007/VirtualClass.git>

### **Branche main**

<https://github.com/AMALK319/virtual-class-microservices-mern.git>

## 6-9. Lien de la vidéo démonstrative



## 6-10. Lien de l'application déployée:

On a deployé le lien la partie frontend, mais faute de temps, on n'a pas pu déployer les microservices de la partie Backend:

<https://ur-virtual-class.herokuapp.com/>



## 7. Conclusion

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète du métier d'ingénieur. En effet, la prise d'initiative, le respect des délais et le travail en équipe seront des aspects essentiels de notre futur métier.

Au final, nous avons beaucoup travaillé pendant les séances et hors des séances. Nous avons compris qu'il est important d'évaluer les connaissances de chacun, et de faire un point sur nos tâches régulièrement.

Ce projet nous a permis de découvrir plusieurs Frameworks et de pratiquer des technologies qui sont les plus utilisées récemment. De plus, nous avons créé un site qui répond à un vrai besoin issue du monde de l'enseignement en ligne qui est devenu primordiale dans notre contexte actuel ,ce qui est très gratifiant.