République Algérienne Démocratique et Populaire
الجمهوريـــة الجزائريـــة الديمقراطيـــة الشعبيـــة
Ministère de L'enseignement Supérieur Et De La Recherche Scientifique
وزارة التعليـــم العالـــي والبحـــث العلمـــي

ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

المدرسة الوطنية العليا للإعلام الآلي
(المعهد الوطني للتكوين في الإعلام الآلي سابقا)
Ecole nationale Supérieure d'Informatique
Ex. INI (Institut National de formation en Informatique)

Rapport du Projet pluridisciplinaire

2ème année Classes Préparatoires (2CP)

PRJP06_EQ10

Thème :

Logiciel interactif de collecte et de développement d'idées pour les projets pédagogiques des étudiants et enseignants de l'ESI

Réalisé par :

Noms & Prénoms	Groupes
Chef d'équipe	
MOSTEFAI Mounir Sofiane	G06
Membres	
KHALEF Badis	G10
BOUGUESSA Wail	G05
CHABBI Ahcen	G10
GUEROUMI Mohamed Lamine Bahaa Eddine	G09
TALEB Mohamed Adnane	G02

Encadré par :

Noms & Prénoms
Mme. KHOURI Selma
Mr. KHOUANI Amin

Année universitaire 2023/2024

Remerciements

Tout d'abord, nous souhaitons exprimer notre profonde reconnaissance envers Dieu qui nous a guidés et soutenus tout au long de ce semestre, malgré les défis imprévus. Sans sa grâce et sa miséricorde, mener à bien ce projet aurait été impossible.

Nous tenons également à remercier chaleureusement toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet. Nos encadreurs, Mme KHOURI et Ms KHOUANI, méritent une reconnaissance spéciale pour leur encadrement exceptionnel, leur soutien constant, leur disponibilité et leurs précieux conseils. Leur expertise et leur expérience ont été une grande source d'inspiration et ont été déterminantes pour la réussite de ce projet.

Nos parents méritent aussi notre gratitude pour leur soutien inconditionnel et leur patience qui nous ont permis de surmonter les moments difficiles et de rester concentrés sur notre objectif. Leur amour et leurs encouragements ont été une force motrice tout au long de cette période.

Enfin, nos amis, proches et enseignants de l'ESI méritent nos remerciements les plus sincères pour leur soutien et leur enthousiasme à chaque étape de ce projet. Leur confiance en nous a été une source d'inspiration constante, et nous sommes fiers de les avoir à nos côtés.

Nous vous remercions tous du fond du cœur.

Table des matières

Remerciements----------2
Table des matières----------3
I. Introduction----------7
II. Analyse du cahier des charges---------- 8
A. Volet ressources---------- 8
B. Volet idéation---------- 8
C. Volet visualisation---------- 8
D. Volet livrables---------- 8
III. Etudes de l'environnement---------- 9
A. Applications similaires----------9
B. Limites des outils existants---------- 9
C. Étude des solutions existantes et comparaison avec le projet---------- 9
IV. Organisation du projet----------10
A. Planification et diagramme de Gantt----------10
1. Phase d'apprentissage et de compréhension du cahier des charges---------- 10
2. Phase de conception----------10
3. Phase de réalisation----------10
4. Phase de finalisation de l'application---------- 10
5. Phase de rédaction---------- 10
B. Outils de gestion de projet----------12
1. Communication---------- 12
2. Collaboration---------- 13
3. Gestion de tâches----------13
C. Gestion des risques----------14
1. Gestion du temps---------- 14
2. L'intégration front et back---------- 14
V. Conception---------- 15
A. Introduction----------15
B. Parcours utilisateur----------15
C. Schéma de la base de données---------- 17
D. Découpage modulaire----------17
1. Authentification---------- 17
2. Gestion de projet---------- 18
3. Panneau administrateur---------- 19
4. Paramètres---------- 20
5. Modules secondaires----------20
E. Diagramme relationnel---------- 20
1. Lien entre le module Authentification et les autres modules (Gestion de projet, Panneau administrateur, Paramètres)---------- 20
2. Lien entre le module Gestion de projet et la Base de données---------- 21
3. Lien entre le module Gestion de projet et le Panneau administrateur----------21
4. Lien entre le module Paramètres et le Panneau administrateur---------- 22

5. Lien entre le module Paramètres et la Base de données----------23
6. Lien entre le Panneau administrateur et la Base de données----------23
7. Lien entre le module Web Scraping et la Base de données---------- 24
F. Charte graphique et identité visuelle---------- 24
1. Logo et nom---------- 24
2. Police de text----------24
3. Palette de couleur----------25
G. Interface utilisateur----------25
1. Wireframe---------- 25
2. Maquette de l'interface utilisateur----------26
● Partie visiteur---------- 26
● Partie utilisateur---------- 26
● Partie administrative---------- 27
● Partie Prototypage----------28
H. Conclusion---------- 28
VI. Réalisation----------29
A. Introduction----------29
B. Outils et technologies utilisés----------29
C. Modules secondaires et scripts---------- 30
1. Initialisation de la base de données---------- 30
2. Chatbot---------- 30
3. Lancement de Docker----------31
D. Le front-end---------- 31
1. Introduction---------- 31
2. Implémentation de Frontend selon la conception----------31
3. Présentation du Front-end---------- 32
E. Le back-end---------- 37
1. Introduction---------- 37
2. Architecture et structure de code---------- 37
Implémentation des APIs----------38
3. Implémentation de la base de données---------- 40
4. Authentification et autorisation----------41
1. Authentification---------- 41
2. Autorisation---------- 41
5. Testes---------- 42
6. Conclusion---------- 42
F. L'Intégration----------43
1. Intégration Frontend-Backend---------- 43
2. Intégration de l'Authentification Google---------- 43
3. Intégration de l'API de Chatbot---------- 43
G. Documentation du logiciel----------44
1. Manuel d'installation----------44
2. Manuel d'utilisation---------- 44
3. Aide en ligne----------44

H. Conclusion---------- 44
VII. Conclusion---------- 45
A. Compétences acquises---------- 45
B. Perspectives---------- 46

I. Introduction

L'étudiant de l'École nationale supérieure d'informatique reçoit plusieurs TD, projets et TP où il doit prendre le temps de penser, clarifier et collecter ses idées avant de commencer la réalisation. De plus, l'étudiant recueille des idées lors des événements scientifiques et artistiques grâce à sa participation active dans différents clubs étudiants.

Généralement, la collecte d'idées n'est pas efficace et plusieurs idées sont perdues ou oubliées, ce qui peut entraîner des conflits. Dans le but de faciliter cette énorme tâche pour les étudiants de l'ESI et de leur offrir un moyen de collecter leurs idées, s'inscrit notre projet qui se déroule durant le second semestre de la deuxième année, avec pour thème la création d'un logiciel interactif de collecte et de développement d'idées pour les projets pédagogiques des étudiants et enseignants de l'ESI.

Toutefois, dans ce rapport, nous commencerons par analyser le cahier des charges qui nous a été transmis par nos encadrants au début de ce semestre, en nous basant sur l'objectif de notre projet et les différentes fonctionnalités demandées. Comme l'organisation et la communication sont les clés de tout travail d'équipe, nous décrirons la gestion de notre équipe ainsi que la planification durant toute la période de travail. Nous présenterons également notre conception de cette application avec tous les détails nécessaires, y compris la maquette représentant une version alpha de notre application.

Enfin, nous aborderons la réalisation où nous mettrons en avant nos choix techniques ainsi que l'implémentation réalisée par chaque sous-équipe : le frontend et le backend, en montrant les résultats obtenus et les perspectives à atteindre.

II. Analyse du cahier des charges

Après avoir reçu le cahier des charges de notre projet, qui consiste à réaliser un logiciel interactif de collecte et de développement d'idées pour les projets pédagogiques des étudiants et enseignants de l'ESI, plus simplement un logiciel où les membres de l'ESI pourront créer des projets d'idéation selon différentes méthodes pour collaborer et collecter leurs idées sur le thème choisi, nous avons commencé par l'analyser afin de le comprendre et d'extraire les tâches à réaliser. Pour cela, nous l'avons divisé en différentes sections de fonctionnalités qui sont les suivantes :

A. Volet ressources

L'utilisateur pourra créer un projet selon les différentes méthodes d'idéation, en particulier les deux méthodes obligatoires à réaliser : le brainstorming et le brainwriting. Il pourra donner un titre, une description, ajouter des collaborateurs, etc. De plus, il pourra lier son projet au club, à un événement ou à un module pédagogique de l'ESI en les recherchant lors de la création du projet.

B. Volet idéation

Une fois que l'utilisateur a terminé de créer son projet, lui et ses collaborateurs pourront passer par toutes les étapes de collecte d'idées. Ils seront ensuite capables de les modifier, de les combiner, de les développer et de les prioriser avant de finaliser le projet. De plus, il sera possible de sauvegarder et de restaurer les projets à partir d'une source externe comme Google Drive.

Il existe une fonctionnalité optionnelle à réaliser, qui consiste à intégrer un chat-bot pour permettre à l'utilisateur d'avoir une assistance personnalisée.

C. Volet visualisation

Notre application doit être intuitive pour faciliter son utilisation. Elle devra proposer une aide en ligne pour guider l'utilisateur vers les fonctionnalités. De plus, elle devra afficher des messages clairs en cas d'anomalie.

D. Volet livrables

Dans le cahier des charges, il nous est demandé de livrer à la fin du projet :

- Ce rapport résumant les travaux et les méthodes que nous avons suivies.
- Une notice d'installation et d'utilisation.
- Une aide en ligne intégrée à notre application.
- L'application elle-même. Nous avons choisi de développer une application web, pour plusieurs raisons et avantages que vous trouverez dans les technologies utilisées dans la partie consacrée à la réalisation. L'un de ces avantages est de viser tout le public de l'ESI et de rendre son utilisation plus facile en ne nécessitant qu'un simple lien sans obligation d'installation sur une machine.

En plus de ces livrables, nous avons été également chargés de créer différents diagrammes que vous trouverez dans les sections suivantes de ce rapport.

En conclusion de cette analyse, la compréhension du cahier des charges a été une étape essentielle pour le bon déroulement de notre projet, nous permettant d'avoir une idée claire sur les tâches à réaliser et sur l'organisation à adopter.

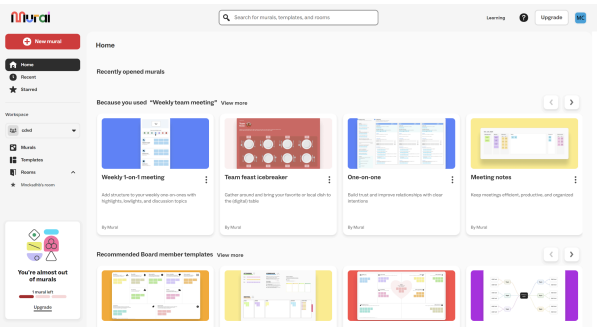
III. Etudes de l'environnement

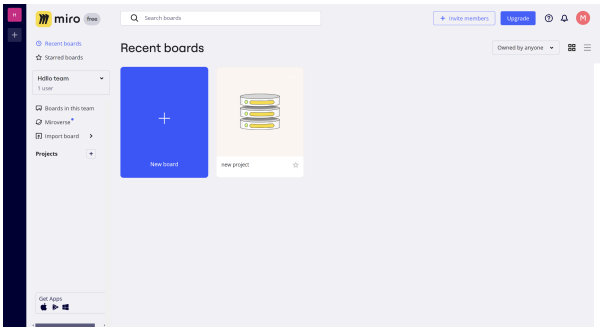
A. Applications similaires

Après avoir reçu le cahier des charges (CDC) et défini la problématique de notre projet, consistant à créer un site web permettant à l'utilisateur de concevoir des projets et des séances d'idéation selon différentes méthodes, de les modifier et de les visualiser à la fin, nous avons entamé une étude de marché pour identifier les applications/sites web offrant des fonctionnalités similaires. Cette étude nous a inspirés pour notre projet et nous a permis d'avoir une idée des fonctionnalités non explicitées par le CDC, ainsi que des fonctionnalités supplémentaires que nous avons jugé utiles pour notre projet.

Les deux principaux sites web sur lesquels nous nous sommes basés pour notre étude sont Miro et Mural, qui permettent à l'utilisateur de créer des whiteboards avec des thèmes différents tels que Brainstorming et Brainwriting. Étant donné que le CDC précisait uniquement que l'application devait offrir la possibilité de créer des séances d'idéations guidées, sans mentions explicites des moyens et des outils nécessaires, nous avons ajouté des fonctionnalités supplémentaires utiles pour aider l'utilisateur à créer facilement ses projet et gérer les séances d'idéation, comme le volet des favorisés, la collaboration en temps réel, l'exportation de la visualisation finale en image, etc.

Nous avons également décidé d'utiliser la structure de Miro comme modèle de structure pour notre application et avons créé une interface utilisateur qui s'inspire fortement de celle-ci.





B. Limites des outils existants

Malheureusement, ces deux sites ne proposaient pas de guide d'utilisation des méthodes d'idéation pour les utilisateurs inexpérimentés. Nous avons donc fini par trouver notre propre solution originale pour cette fonctionnalité principale et l'avons présentée aux encadrants pour validation avant de commencer sa réalisation.

C. Étude des solutions existantes et comparaison avec le projet

Il y avait également certaines fonctionnalités mentionnées dans le CDC, mais sans explication suffisante sur leur fonctionnement et leurs limites. Nous avons donc utilisé encore une fois MIRO et Mural comme référence et avons créé ces fonctionnalités en nous basant sur

la façon dont elles sont implémentées sur ces deux sites web, par exemple la gestion des projets après création.

IV. Organisation du projet

A. Planification et diagramme de Gantt

Après avoir reçu le cahier des charges et discuté de chaque partie de la problématique, nous avons établi un planning pour organiser efficacement le travail sur le projet. Ce processus nous a permis de convenir d'une répartition en 5 phases principales :

1. Phase d'apprentissage et de compréhension du cahier des charges

Nous avons consacré deux semaines à la lecture approfondie et à l'analyse du cahier des charges pour mieux comprendre la problématique, les outils imposés, et étudier le marché. Cette phase nous a également permis de nous documenter sur les outils à utiliser et de combler nos lacunes.

2. Phase de conception

Après analyse du cahier des charges, nous avons proposé des solutions aux problèmes identifiés, défini les fonctionnalités requises et découpé le projet en tâches attribuées à chaque membre de l'équipe. Nous avons élaboré le diagramme de classe représentant la structure de l'application et le diagramme de flux de travail représentant l'expérience utilisateur. De plus, nous avons finalisé l'identité visuelle de l'application, y compris le logo, le nom et les palettes de couleurs.

3. Phase de réalisation

Après validation de la conception et du design, nous sommes passés à la phase de développement. Nous avons commencé par implémenter les fonctionnalités les plus importantes, en progressant vers les moins critiques, en assurant que le front-end et le back-end étaient prêts avant leur intégration et les tests.

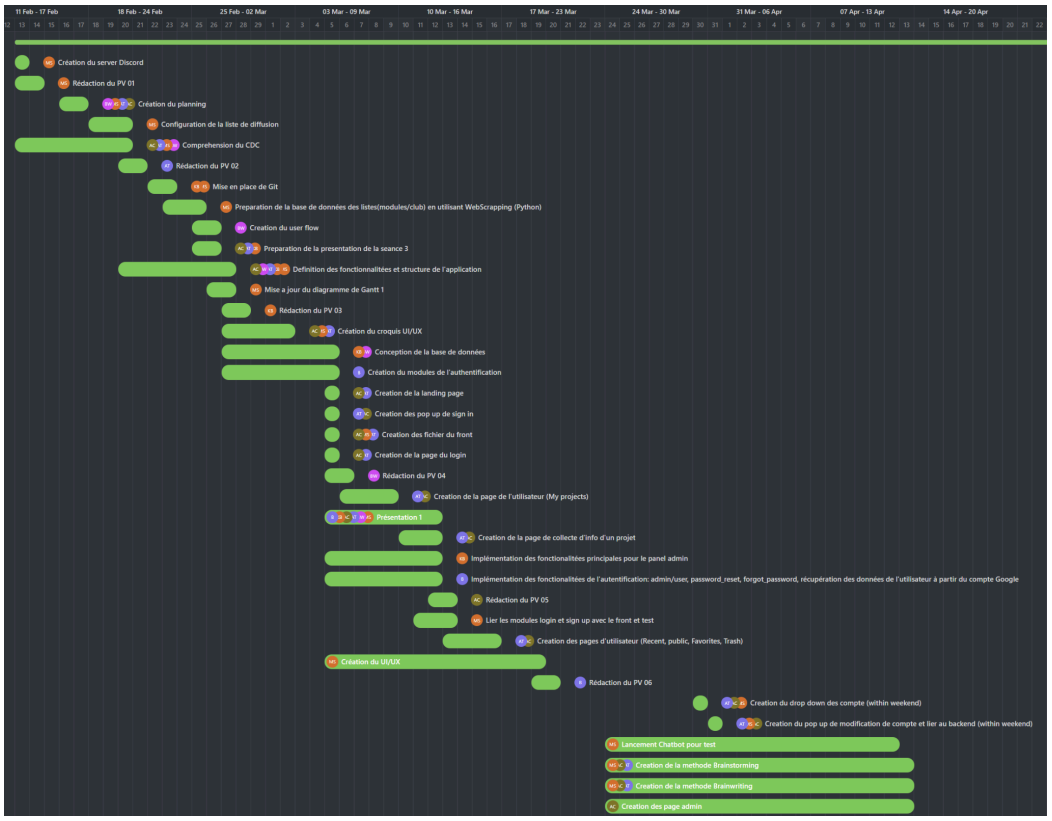
4. Phase de finalisation de l'application

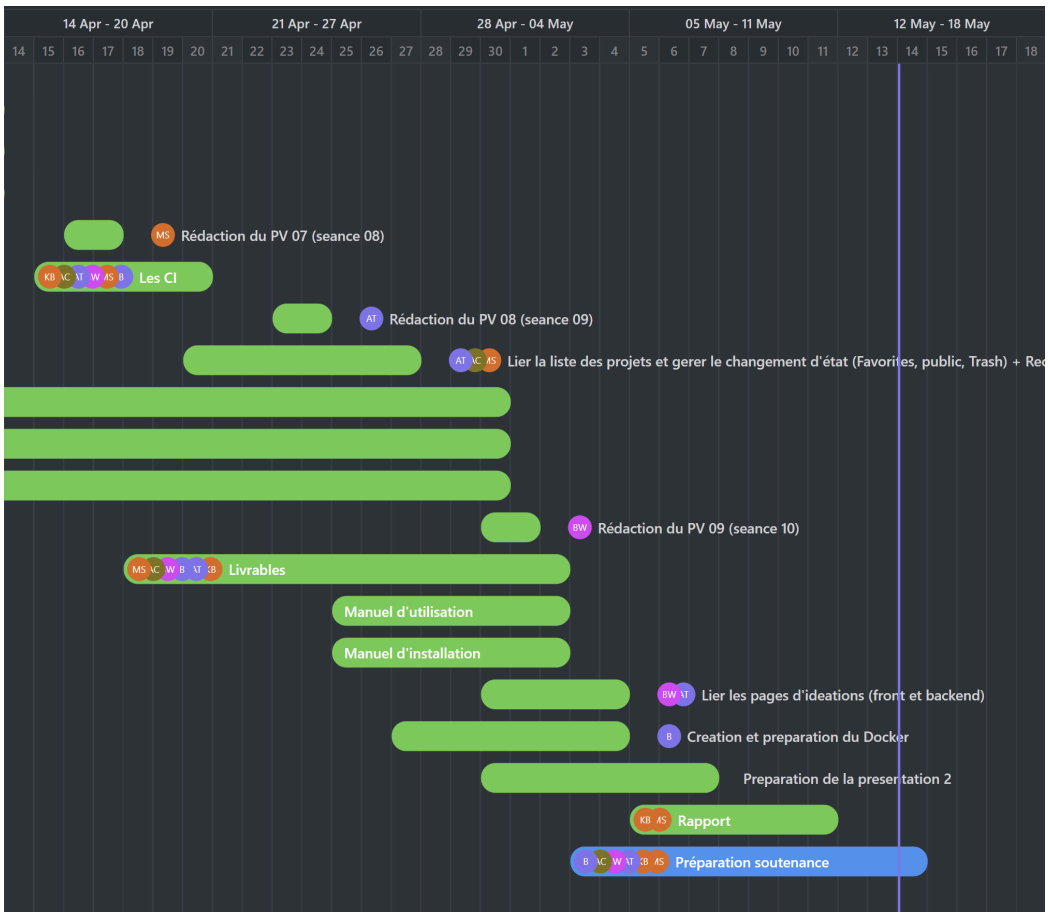
À mesure que la phase de réalisation touchait à sa fin, nous avons préparé la version finale de l'application en nettoyant le code des messages de test.

5. Phase de rédaction

Cette phase consiste à rédiger le rapport à partir des étapes de conception validées par nos superviseurs lors de la première évaluation. Nous avons prévu de structurer le rapport en différentes parties, chacune correspondant à une étape spécifique du projet.

Cette planification détaillée nous a permis de suivre efficacement notre progression et de garantir la réalisation réussie de chaque étape du projet.


11 Feb - 17 Feb
18 Feb - 24 Feb
25 Feb - 02 Mar
03 Mar - 09 Mar
10 Mar - 16 Mar
17 Mar - 23 Mar
24 Mar - 30 Mar
31 Mar - 06 Apr
07 Apr - 13 Apr
14 Apr - 20 Apr
Création du server Discord
Rédaction du PV 01
Création du planning
Configuration de la liste de diffusion
Comprehension du CDC
Rédaction du PV 02
Mise en place de Git
Preparation de la base de données des listes(modules/club) en utilisant WebScrapping (Python)
Creation du user flow
Preparation de la presentation de la seance 3
Definition des fonctionnalitées et structure de l'application
Mise a jour du diagramme de Gantt 1
Rédaction du PV 03
Creation du croquis UI/UX
Conception de la base de données
Création du modules de l'authentification
Creation de la landing page
Creation des pop up de sign in
Creation des fichier du front
Creation de la page du login
Rédaction du PV 04
Creation de la page de l'utilisateur (My projects)
Présentation 1
Creation de la page de collecte d'info d'un projet
Implémentation des fonctionalitées principales pour le panel admin
Implémentation des fonctionalitées de l'autentification: admin/user, password_reset, forgot_password, récupération des données de l'utilisateur à partir du compte Google
Rédaction du PV 05
Lier les modules login et sign up avec le front et test
Creation des pages d'utilisateur (Recent, public, Favorites, Trash)
Création du UI/UX
Rédaction du PV 06
Creation du drop down des compte (within weekend)
Creation du pop up de modification de compte et lier au backend (within weekend)
Lancement Chatbot pour test
Creation de la methode Brainstorming
Creation de la methode Brainwriting
Creation des page admin

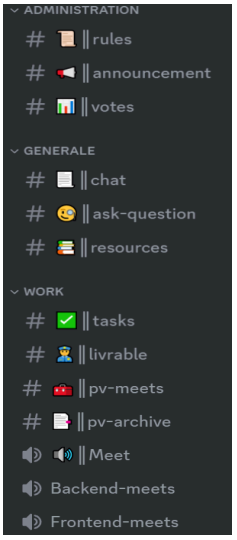

14 Apr - 20 Apr
21 Apr - 27 Apr
28 Apr - 04 May
05 May - 11 May
12 May - 18 May
Rédaction du PV 07 (seance 08)
Les CI
Rédaction du PV 08 (seance 09)
Lier la liste des projets et gerer le changement d'état (Favorites, public, Trash) + Rec
Rédaction du PV 09 (seance 10)
Livrables
Manuel d'utilisation
Manuel d'installation
Lier les pages d'ideations (front et backend)
Creation et preparation du Docker
Preparation de la presentation 2
Rapport
Préparation soutenance

B. Outils de gestion de projet

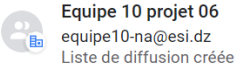
Dans le but d'optimiser notre organisation et de maximiser l'efficacité, nous avons mis en place une panoplie d'outils de gestion de projet, que nous pouvons subdiviser en trois catégories principales :

1. Communication

Discord est une plateforme de communication en ligne initialement conçue pour les joueurs, mais qui s'est étendue à de nombreuses autres communautés. Elle permet des discussions textuelles et vocales en temps réel, ainsi que le partage de fichiers et de médias.
Nous avons utilisé Discord en créant notre propre serveur dédié, sur lequel nous avons centralisé toutes les informations relatives au projet. Cela inclut le partage de ressources pertinentes, les discussions autour du travail en cours, ainsi que la planification et la coordination des réunions, des sessions de travail et des discussions. En utilisant les fonctionnalités de canaux dédiés et de permissions spécifiques, nous avons pu organiser efficacement nos échanges et assurer une communication fluide et ciblée.



Gmail est un service de messagerie électronique développé par Google. Dans le cadre de notre projet, nous avons créé une liste de diffusion via Gmail, qui regroupe tous les membres de l'équipe. Cette liste nous permet de communiquer efficacement avec l'ensemble du groupe. Nous pouvons partager des mises à jour, des documents importants, des annonces et coordonner les actions de manière rapide et pratique.



Google Meet est une plateforme de visioconférence développée par Google, utilisée pour nos réunions de projet. Cela nous a permis de nous réunir de manière simple et efficace, favorisant des sessions de travail productives et une communication transparente au sein de l'équipe, grâce à ses fonctionnalités conviviales et sa compatibilité avec d'autres outils Google.

2. Collaboration

Figma est une plateforme de conception collaborative en ligne, offrant des outils puissants pour créer des interfaces utilisateur et des prototypes. Nous avons exploité ses fonctionnalités pour élaborer et partager efficacement nos designs. Plus spécifiquement, nous avons utilisé son outil **Figma Jam** pour créer des diagrammes, en particulier le diagramme user flow, permettant ainsi une visualisation claire des parcours utilisateurs dans notre projet.

Lucidchart est une plateforme intuitive de création de diagrammes en ligne, offrant une gamme d'outils pour représenter graphiquement des données et des relations complexes. Nous avons utilisé Lucidchart pour concevoir notre diagramme relationnel, qui a servi à modéliser en détail les liens et les interactions entre les différentes entités du projet. Grâce à ses fonctionnalités avancées, nous avons pu créer un diagramme clair et précis, offrant une compréhension approfondie de la structure et des interrelations du système.

Google Docs est une suite bureautique en ligne, offrant des fonctionnalités de traitement de texte, de création de feuilles de calcul et de préparation de présentations. Nous avons utilisé Google Docs pour rédiger et collaborer sur divers documents de projet, tels que les propositions, les rapports d'avancement et les spécifications techniques. De plus, nous avons également employé Google Docs pour la rédaction et la conservation des procès-verbaux de réunions, permettant ainsi une documentation précise et accessible des discussions et des décisions prises lors de nos rencontres. Grâce à sa fonction de partage en temps réel, nous avons pu travailler de manière collaborative et maintenir l'historique de nos activités et de nos échanges au sein de l'équipe.

Canva est une plateforme de conception graphique en ligne, qui offre une multitude d'outils et de modèles pour créer des designs visuels attractifs. Nous l'avons utilisé pour concevoir des éléments graphiques comme les illustrations, ainsi que pour créer nos présentations, ce qui a rendu notre projet plus attractif visuellement.

3. Gestion de tâches

ClickUp est une plateforme de gestion de projet en ligne qui offre une variété d'outils pour planifier, organiser et suivre les projets. Nous avons utilisé ClickUp pour créer et gérer notre diagramme de Gantt, permettant une visualisation claire des dépendances et des planifications des tâches. Nous avons également utilisé ses fonctionnalités pour répartir les tâches entre les membres de l'équipe et suivre leur progression en temps réel, ce qui nous a permis de

maintenir une bonne coordination et de respecter les délais du projet.

MatchWare MindView est un logiciel de mind mapping et de gestion de projet qui offre des outils pour organiser visuellement les idées et planifier les projets. Au début de notre projet, nous avons utilisé MindView pour créer notre diagramme de Gantt, profitant de ses fonctionnalités pour représenter les dépendances et les échéances des tâches. Cependant, nous avons ultérieurement opté pour ClickUp, en raison de sa richesse de fonctionnalités et de son interface plus intuitive, ce qui nous a permis de mieux gérer la répartition des tâches et leur suivi.

Google Tasks est une application de gestion des tâches légère et simple, intégrée à l'écosystème Google. Nous l'avons principalement utilisée pour gérer des tâches secondaires et des activités ponctuelles tout au long du projet. Bien que moins complexe que d'autres outils de gestion de projet, sa simplicité d'utilisation en fait un choix pratique pour suivre les petites tâches et les rappels quotidiens, complétant ainsi notre gestion globale des activités.

C. Gestion des risques

La gestion des risques dans un projet est une composante essentielle visant à identifier, évaluer et atténuer les facteurs susceptibles d'entraver la réalisation des objectifs. Dans cette section, nous détaillerons les risques anticipés et les stratégies mises en place pour les gérer efficacement dans le cadre de notre projet. Nous examinerons notamment deux aspects cruciaux : la gestion du temps et l'intégration front et back.

1. Gestion du temps

Dans le cadre d'un projet s'étalant sur 10 semaines et impliquant une équipe relativement novice dans les technologies utilisées, la planification du temps est primordiale. Ainsi, nous avons élaboré un planning de Gantt détaillé, définissant précisément les tâches et leur répartition entre les membres de l'équipe. Pour assurer une prise en main efficace et une meilleure productivité, nous avons consacré deux semaines initiales à l'apprentissage des outils et technologies nécessaires (Git, Typescript, Express.js, React, Tailwind, HTML, CSS, Figma ...). Nous avons également anticipé les interruptions potentielles telles que les semaines de contrôles, les fêtes religieuses (Aïd) et les vacances, en réservant des périodes spécifiques pour ces occasions. De plus, nous avons inclus deux semaines supplémentaires pour faire face à d'éventuels retards ou imprévus, ce qui s'est avéré extrêmement utile.

2. L’intégration front et back

L'intégration des composants front-end et back-end constitue souvent un défi majeur, en particulier pour une équipe de débutants. Ainsi, nous avons accordé une attention particulière à cette phase du projet, en allouant un temps considérable pour réaliser cette intégration de manière rigoureuse. Nous avons mis en place des sessions de travail dédiées à cette étape critique tout au long du projet, permettant ainsi de résoudre rapidement les éventuels problèmes rencontrés et d'assurer un fonctionnement cohérent de l'ensemble du système.

V. Conception

A. Introduction

La phase de conception est le noyau de tout projet, c’est une étape clé pour sa réussite qui permet de clarifier ses objectifs, de définir les fonctionnalités et de sélectionner les différentes options du traitement de la problématique tout en se basant sur l’analyse du cahier des charges. Le but principal de cette phase est de s’assurer que le produit final satisfait les besoins des utilisateurs, qu’il est techniquement réalisable et qu’il respecte les contraintes temporelles.

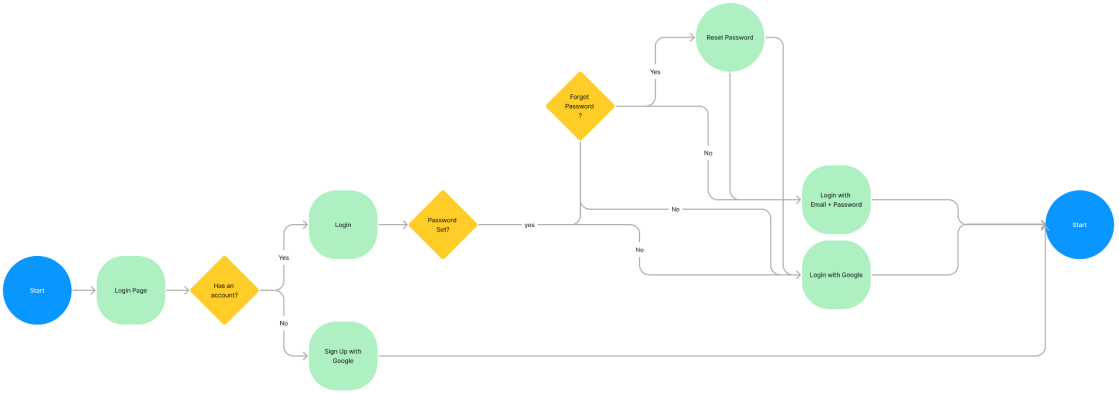
B. Parcours utilisateur

Dans le domaine de la conception de projets, le user flow, également connu sous le nom de parcours d’utilisateur, joue un rôle crucial dans la création d'expériences utilisateur optimales et dans le succès global du projet. Le user flow représente les étapes et actions qu'un utilisateur effectue pour atteindre un objectif spécifique sur l’application. Il s'agit d'un diagramme décrivant le chemin emprunté par l'utilisateur et ses interactions avec le système à chaque étape du processus.

Pour structurer efficacement notre application, nous avons décidé de diviser les actions en trois types en fonction du rôle de l’utilisateur :

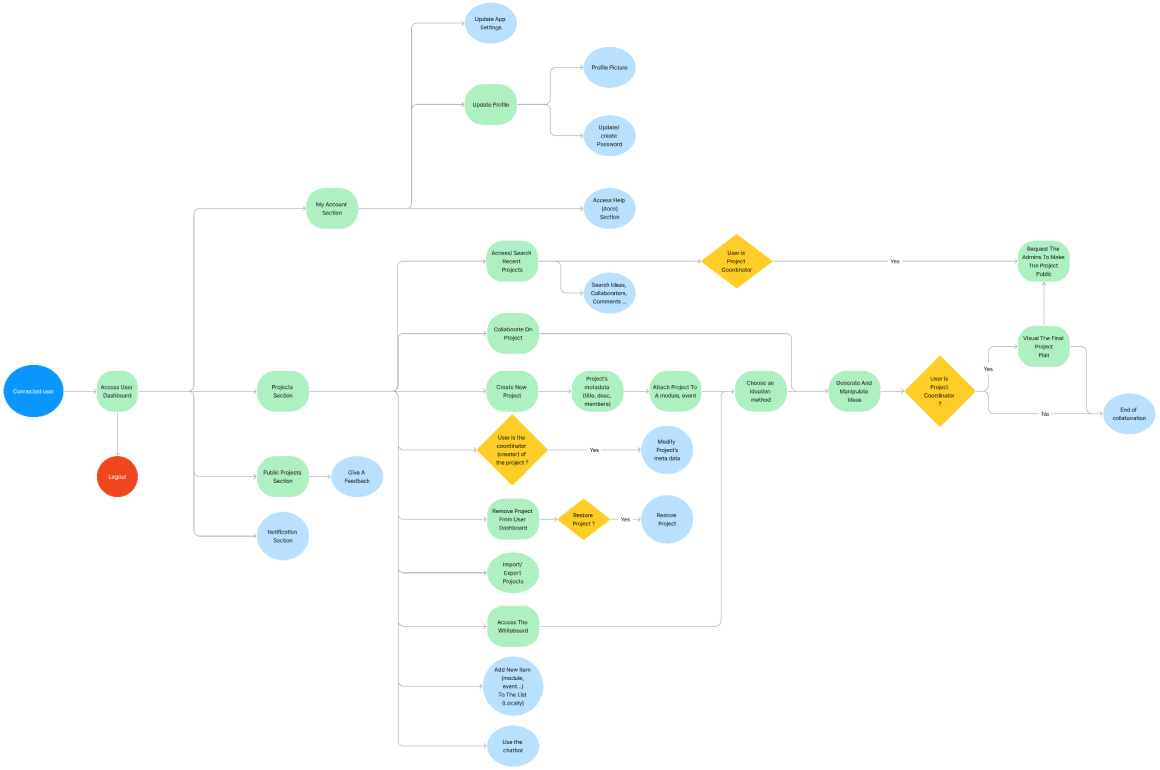
- **Visiteur ou Utilisateur non connecté :**

Les nouveaux utilisateurs ou visiteurs peuvent voir la page d’accueil et se connecter à leur compte s'ils en ont un, ou créer un nouveau compte s'ils le souhaitent.



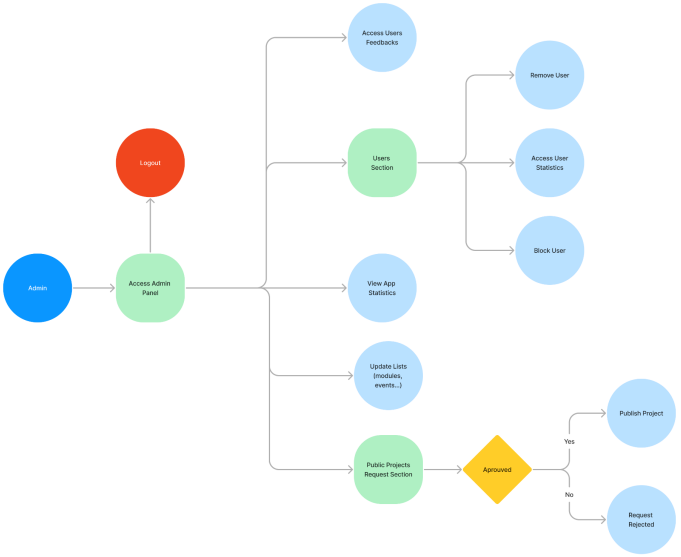
- **Utilisateur connecté :**

Une fois l'authentification et la connexion réussies, l’utilisateur a accès à la page d’accueil de l’application où il peut consulter tous les projets qu'il a créé. Il peut également modifier, supprimer ou créer de nouveaux projets, ainsi que modifier les paramètres de son compte.



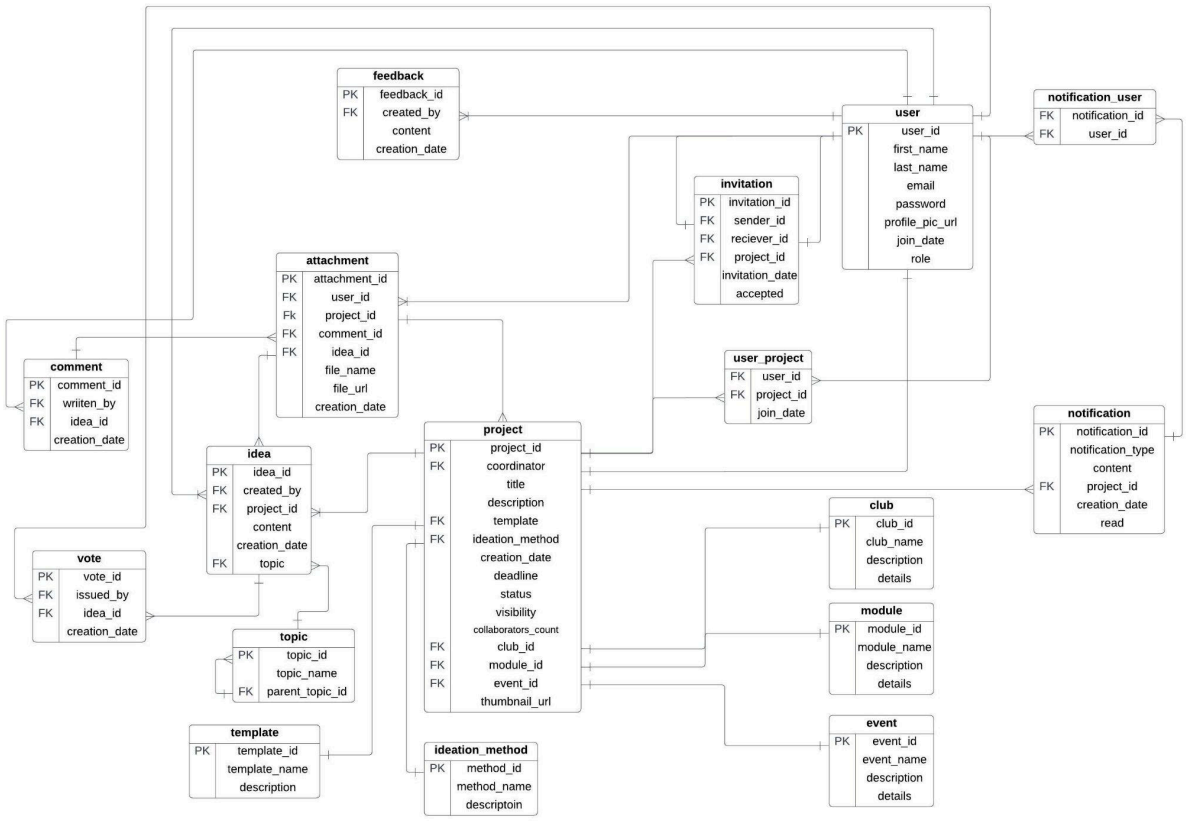
- **Administrateur :**

Un administrateur est considéré comme un utilisateur, mais il bénéficie de privilèges supplémentaires qui lui permettent d'accéder à la page d'administration. Là, il peut consulter différentes statistiques du site et accéder à d'autres fonctionnalités réservées aux administrateurs.



C. Schéma de la base de données

Le schéma de la base de données désigne la représentation complète des entités, des relations et des attributs qui composent le système de gestion de base de données de notre application. Il offre une vue d'ensemble claire de la manière dont les données sont organisées et interconnectées au sein du système.

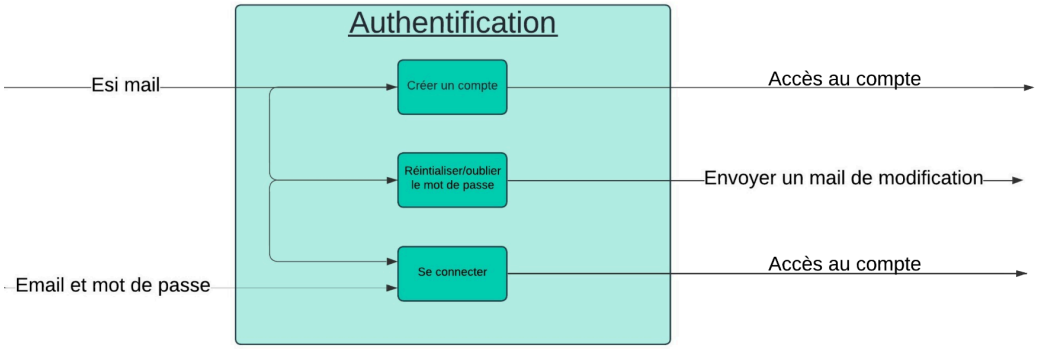


D. Découpage modulaire

Le découpage modulaire d'un logiciel est une approche de conception qui consiste à diviser le système en modules autonomes et interconnectés, chacun responsable d'une fonctionnalité spécifique. Cela permet une meilleure organisation, une maintenance plus facile et une meilleure capacité à faire évoluer le logiciel. Pour découper notre logiciel de manière modulaire, nous avons d'abord analysé minutieusement le cahier des charges et défini toutes les fonctionnalités lors de la conception. Nous avons ensuite opté pour une approche de regroupement en systèmes, puis en sur-systèmes. Cela nous a permis de créer quatre modules globaux : l'Authentification, la Gestion de projet, le Panneau administrateur et les Paramètres. En plus de ces modules principaux, nous avons également développé d'autres modules secondaires et indépendants, tels que le Web Scraping et le Chatbot.

1. Authentification

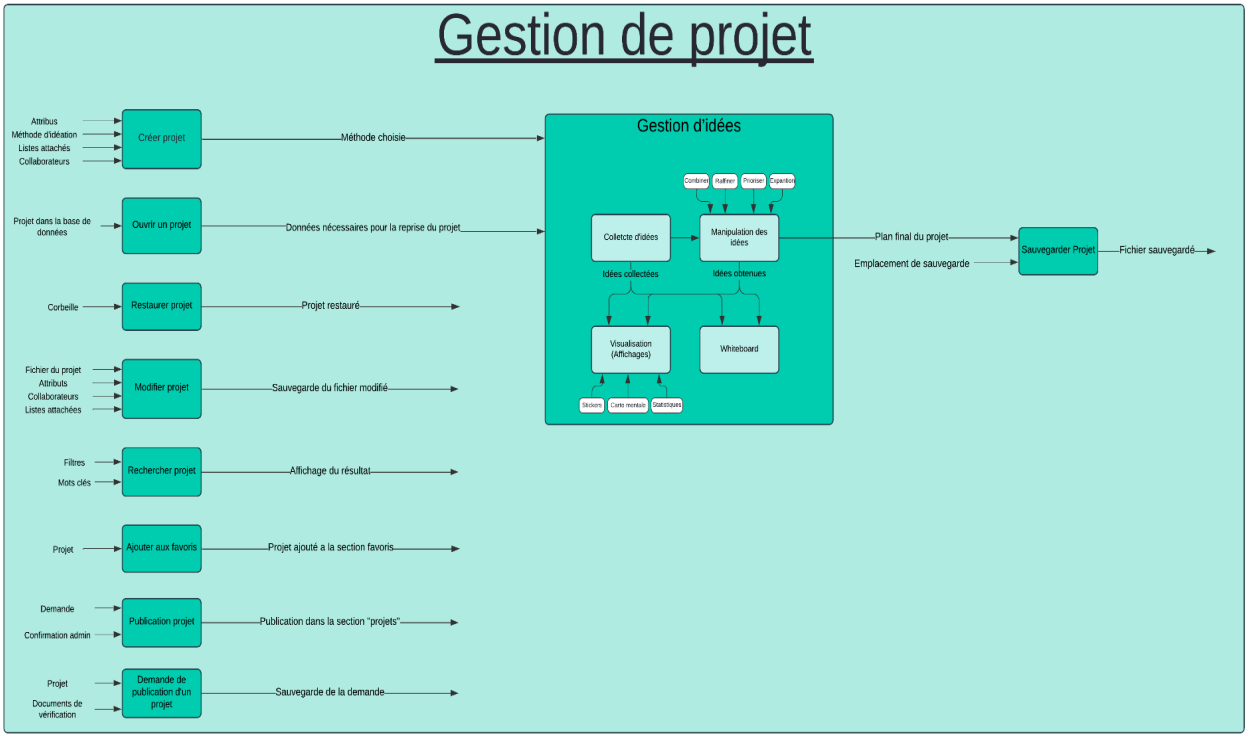
Le module Authentification permet à l'utilisateur de créer un compte en utilisant son adresse e-mail de l'ESI, de se connecter et de réinitialiser son mot de passe. En entrée, il prend l'adresse e-mail pour la connexion Google ou l'e-mail et le mot de passe, et en sortie, il donne accès au compte utilisateur et donc à tous les autres modules.



2. Gestion de projet

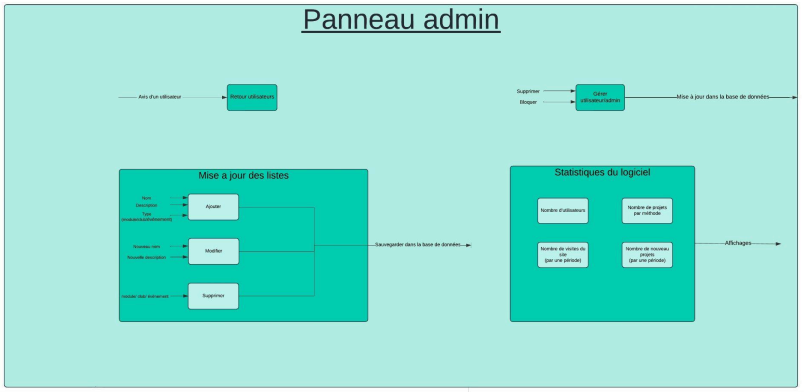
Le module Gestion de projet permet à l'utilisateur de gérer tous les aspects liés à un projet, notamment sa création, sa sauvegarde, son ouverture, sa modification et sa recherche. Il inclut également le module principal de notre projet : la gestion des idées. Cette fonctionnalité comprend la collecte, la manipulation (combinaison, raffinement, priorisation, expansion) et l'affichage des idées (sous forme de stickers), ainsi que la fonctionnalité whiteboard.

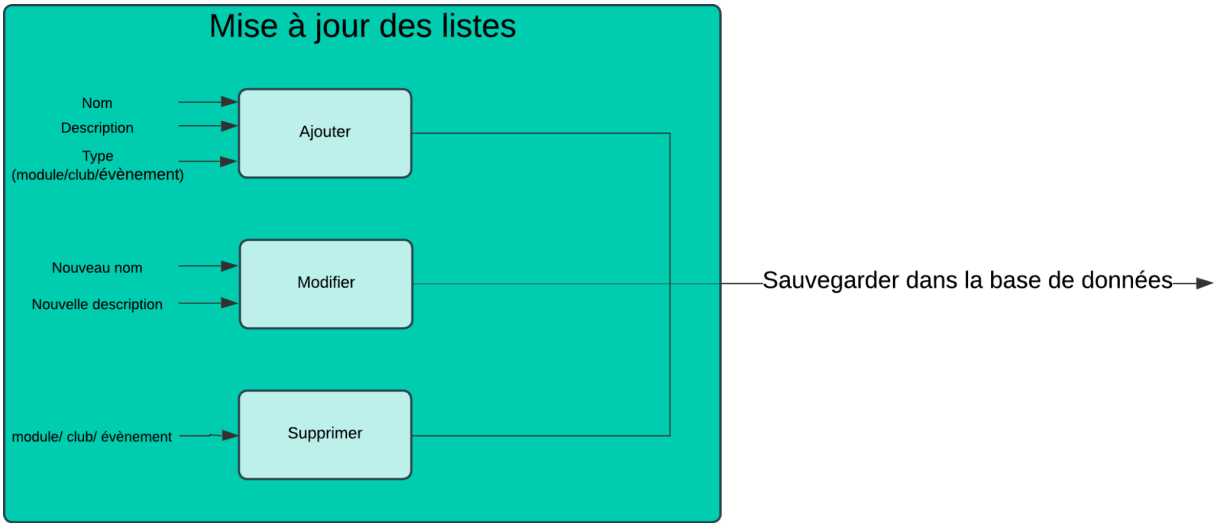
De plus, ce module propose plusieurs fonctionnalités supplémentaires. Les utilisateurs peuvent ajouter des projets à leurs favoris, accéder à la corbeille pour restaurer des projets supprimés et consulter la liste des projets publics, qui comprend les projets relevant de l'école approuvés par l'administrateur. Tous les utilisateurs ont la possibilité de soumettre une demande de publication d'un projet pour qu'il soit rendu public.

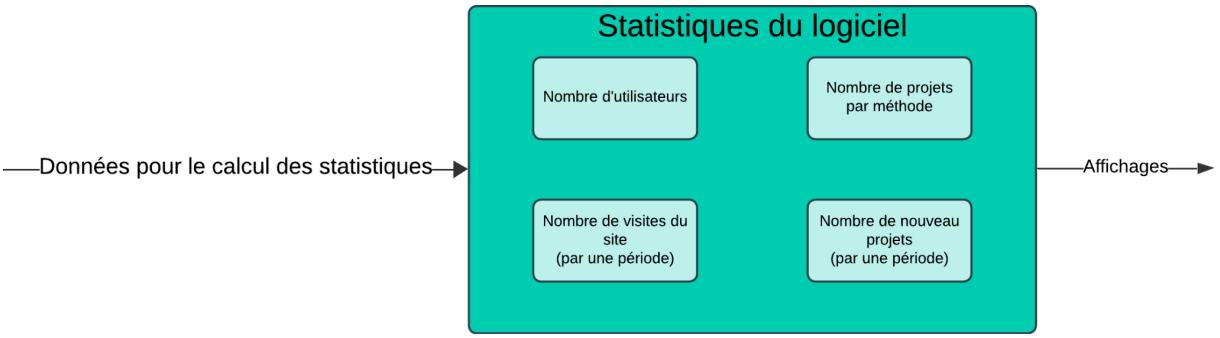


3. Panneau administrateur

Le module Panneau administrateur permet aux admins de gérer les utilisateurs en leur offrant la possibilité de les supprimer ou de les bloquer. Il facilite également la mise à jour des listes de modules, clubs et événements, tout en fournissant un accès aux statistiques du logiciel telles que le nombre d'utilisateurs, le nombre de projets par méthode, ainsi que les visites du site et les nouveaux projets des dernières 24 heures. De plus, il permet de consulter les retours des utilisateurs sur le logiciel et d'approuver les demandes de publication de projets.

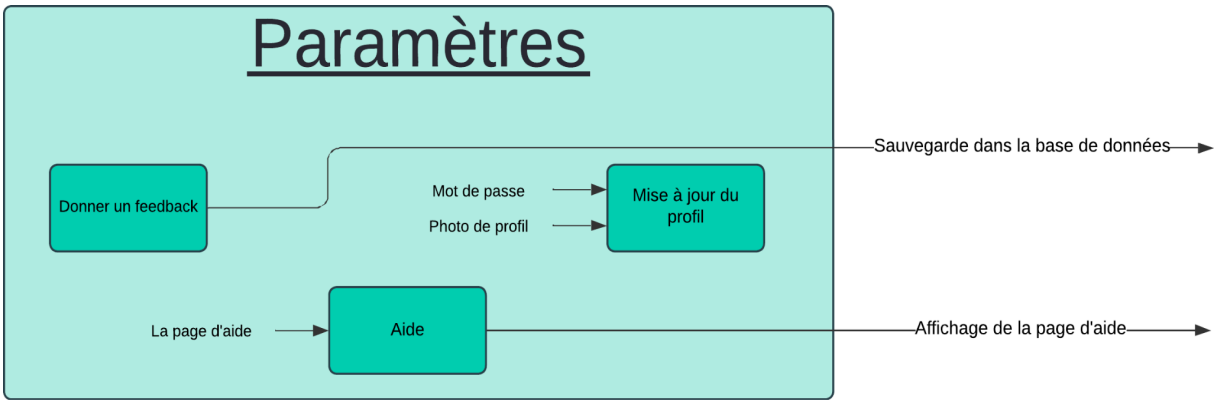






4. Paramètres

Le module Paramètres offre à l'utilisateur la possibilité de mettre à jour son profil en changeant sa photo ou son mot de passe. De plus, il lui permet d'accéder à l'aide du logiciel et de fournir des retours sur le logiciel.



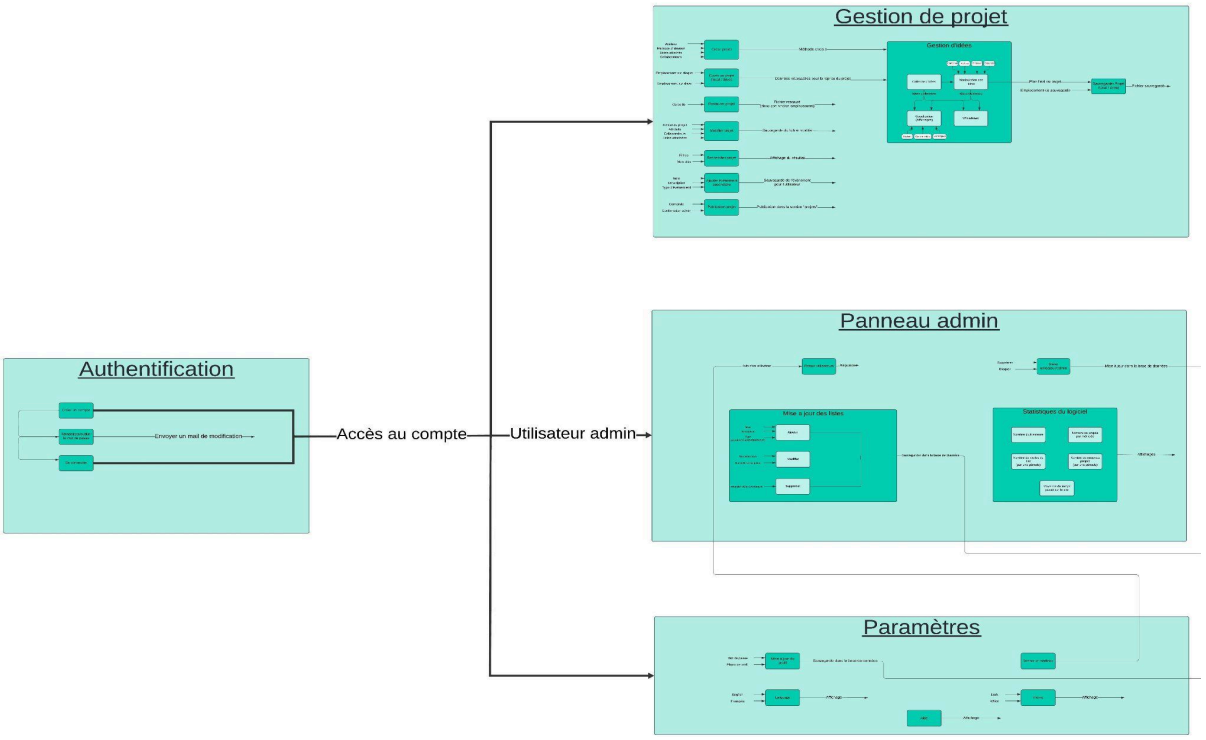
5. Modules secondaires

E. Diagramme relationnel

Le diagramme relationnel est une représentation graphique utilisée pour illustrer les interactions et les dépendances entre les différentes entités d'un système. Il permet de visualiser les relations fonctionnelles entre les composants, facilitant ainsi la compréhension de la structure et du fonctionnement interne d'un projet. Dans cette section, nous examinerons de près chaque élément de notre diagramme relationnel, en analysant les liens essentiels.

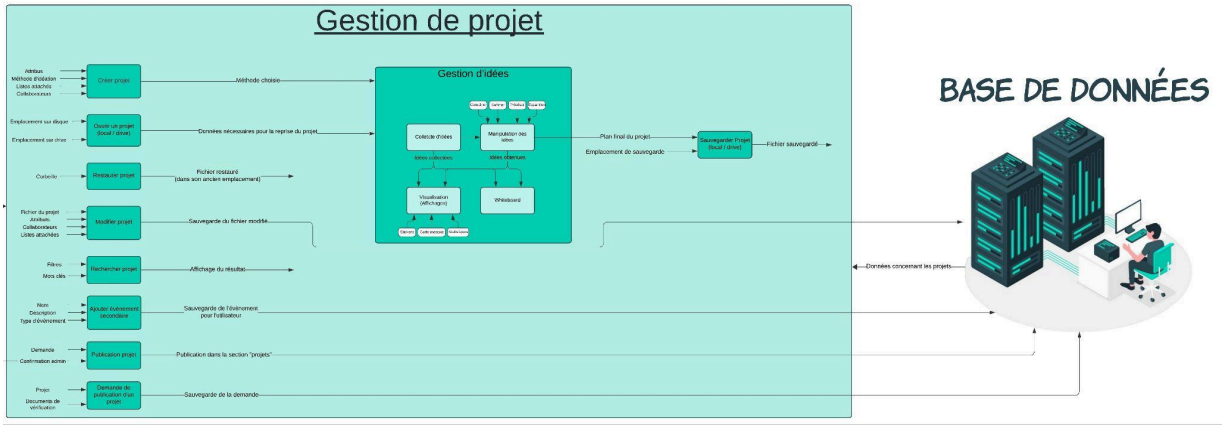
1. Lien entre le module Authentification et les autres modules (Gestion de projet, Panneau administrateur, Paramètres)

Le module d'Authentification joue un rôle crucial en fournissant l'accès aux autres modules du système. Cette autorisation est essentielle pour garantir la sécurité et la confidentialité des données.



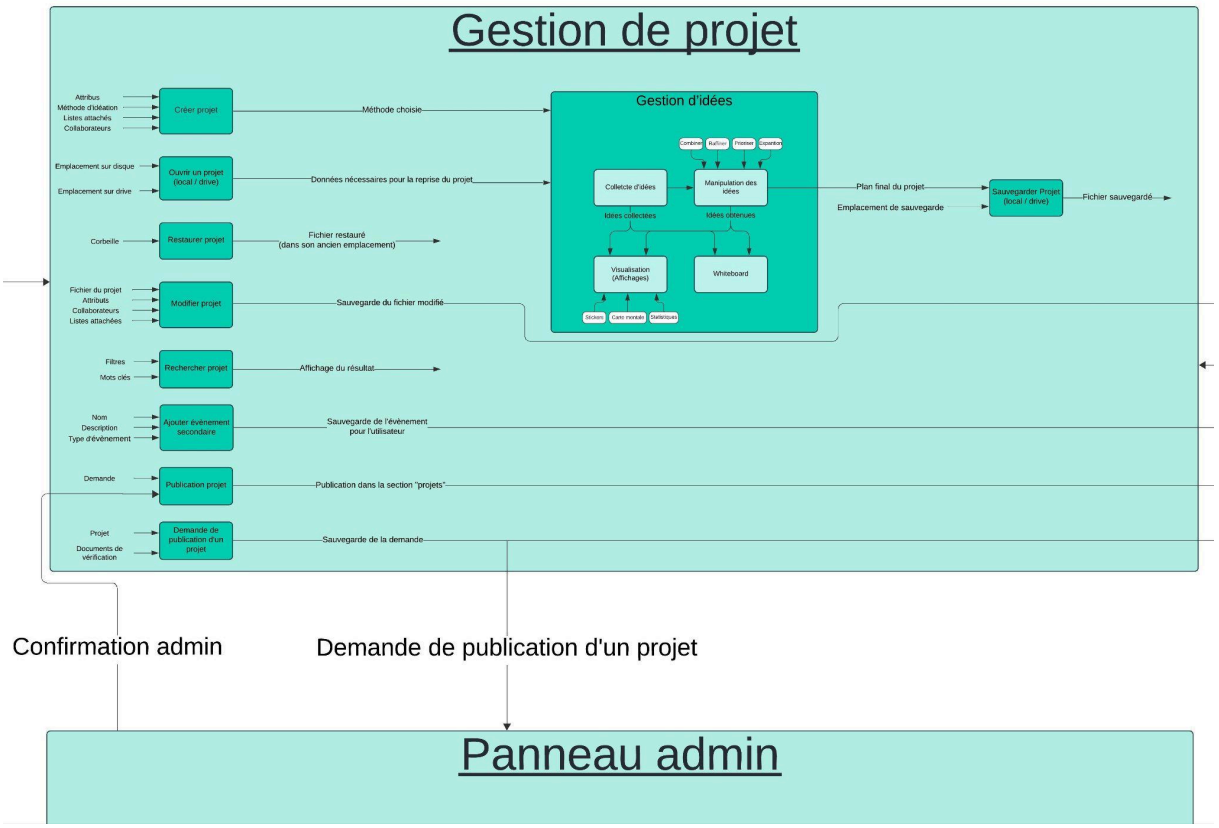
2. Lien entre le module Gestion de projet et la Base de données

Le module de Gestion de projet interagit fortement avec la Base de données pour sauvegarder toutes les données relatives aux projets, effectuer des modifications et récupérer les informations nécessaires concernant les projets existants.



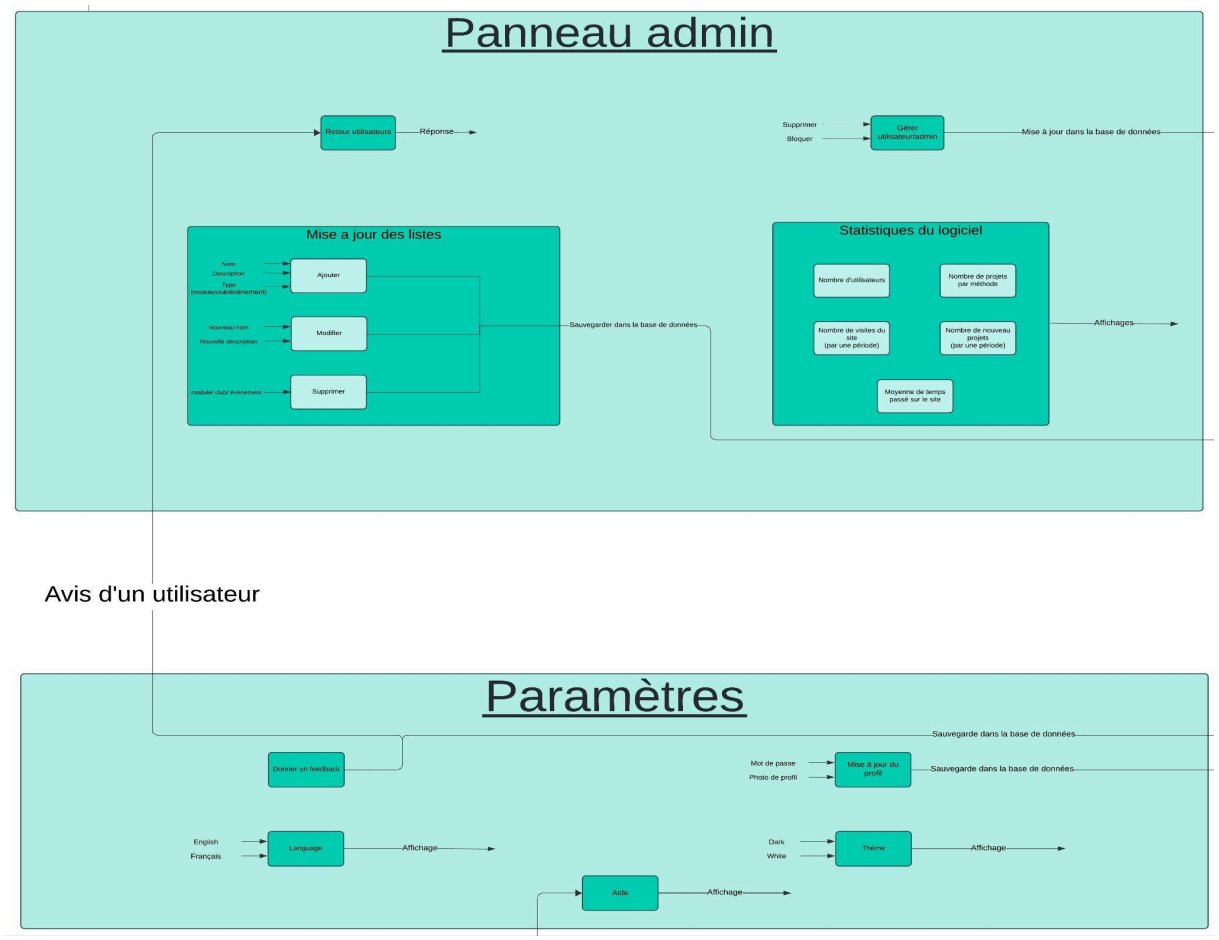
3. Lien entre le module Gestion de projet et le Panneau administrateur

Le module de Gestion de projet communique avec le Panneau administrateur en émettant des demandes de publication de projets et en recevant des confirmations de publication, ce qui assure une gestion efficace de la fonctionnalité “Projets publics”.



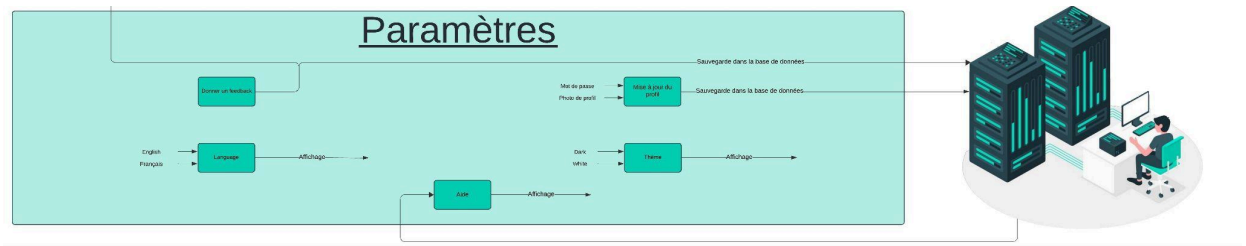
4. Lien entre le module Paramètres et le Panneau administrateur

Le module Paramètres envoie des feedbacks utilisateurs vers le Panneau d'admin pour que ce dernier puisse y répondre, permettant ainsi une communication bidirectionnelle pour améliorer en continu l'expérience utilisateur.



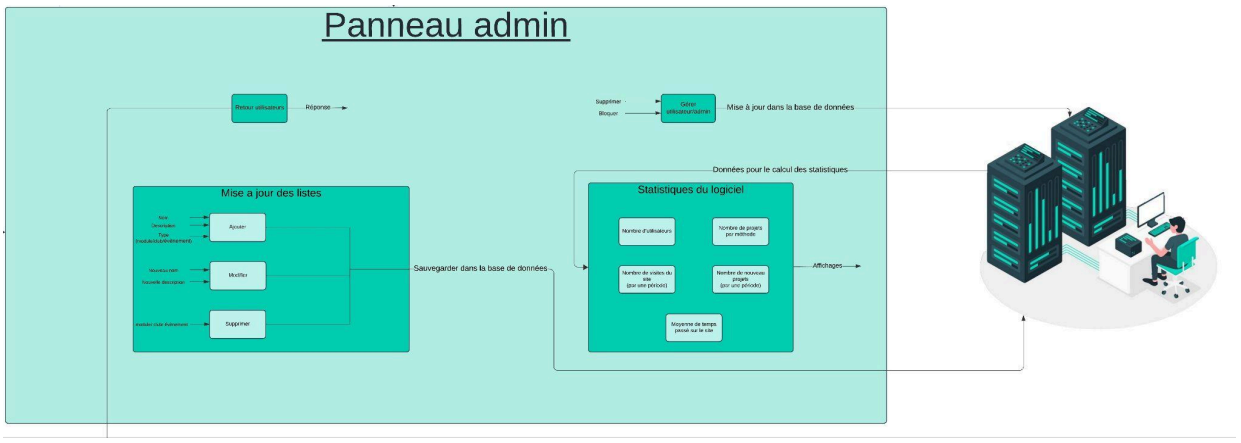
5. Lien entre le module Paramètres et la Base de données

Le module Paramètres interagit avec la Base de données pour sauvegarder et modifier les informations utilisateur telles que la photo de profil et le mot de passe, ainsi que pour sauvegarder les feedbacks et importer l'aide en ligne.



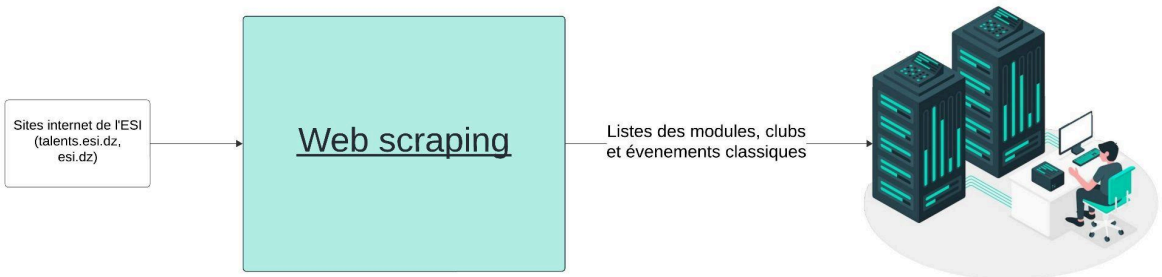
6. Lien entre le Panneau administrateur et la Base de données

Le Panneau administrateur effectue diverses opérations dans la Base de données, telles que la suppression et ban des utilisateurs, la sauvegarde des listes de modules, clubs et événements, la gestion des réponses aux feedbacks utilisateurs, ainsi que l'importation des données nécessaires depuis la base de données pour le calcul des statistiques.



7. Lien entre le module Web Scraping et la Base de données

Le module de Web Scraping collecte des données à partir des sites internet de l'ESI concernant les listes de modules, clubs et événements, et les sauvegarde dans la Base de données.



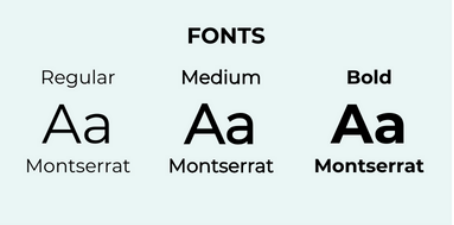
F. Charte graphique et identité visuelle

1. Logo et nom

Notre application se nomme “ESIDEA”, un nom qui résulte de la combinaison des mots “ESI” et “idea”. Le terme “ESI” est le nom de notre école et aussi l'établissement à qui notre application est destinée , tandis que “idea” représente l'idée en anglais. Nous avons choisi de fusionner ces deux mots pour créer un nom distinctif.

Comme montré dans l’image à côté, le logo est constitué du nom de notre application. Au centre, une lampe a été ajoutée sur la lettre “I”, symbolisant une idée. Cette représentation graphique, simple, efficace et distinctive met en avant la nature même de notre application, qui est de faciliter la création et l'organisation des idées à travers des séances d'idéations.

2. Police de text

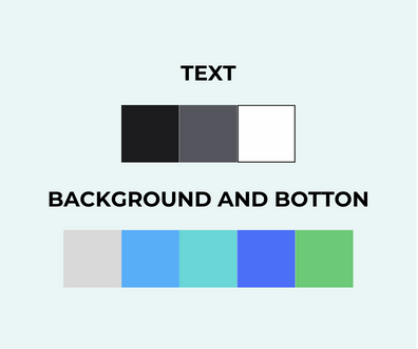


Les polices dans une application qui va être utilisée par un grand nombre de personnes doivent être lisibles, simples et minimalistes. Et puisque nous faisons partie

du monde des informaticiens, la modernité est aussi très importante. Pour cela, nous avons opté pour la police ‘sans-serif’ qui regroupe toutes les propriétés que nous avons cité auparavant et qui est “Montserrat”.

3. Palette de couleur

Les couleurs sont également importantes voir plus, car chaque couleur provoque instantanément une réaction émotionnelle chez l’utilisateur. Ce qui nous a amené à choisir une palette de couleurs très riche où chacune d’elles fera véhiculer un message et un but à travers l’écran.



les différentes couleurs que nous avons adopté représentent:

Le gris : la technologie, l’intelligence et la formalité.

Le gris clair : Simplicité, neutralité, et clarté dans la conception visuelle.

Le bleu : la confiance, la sécurité et le professionnalisme.

Le bleu clair : calme, sérénité, et harmonie.

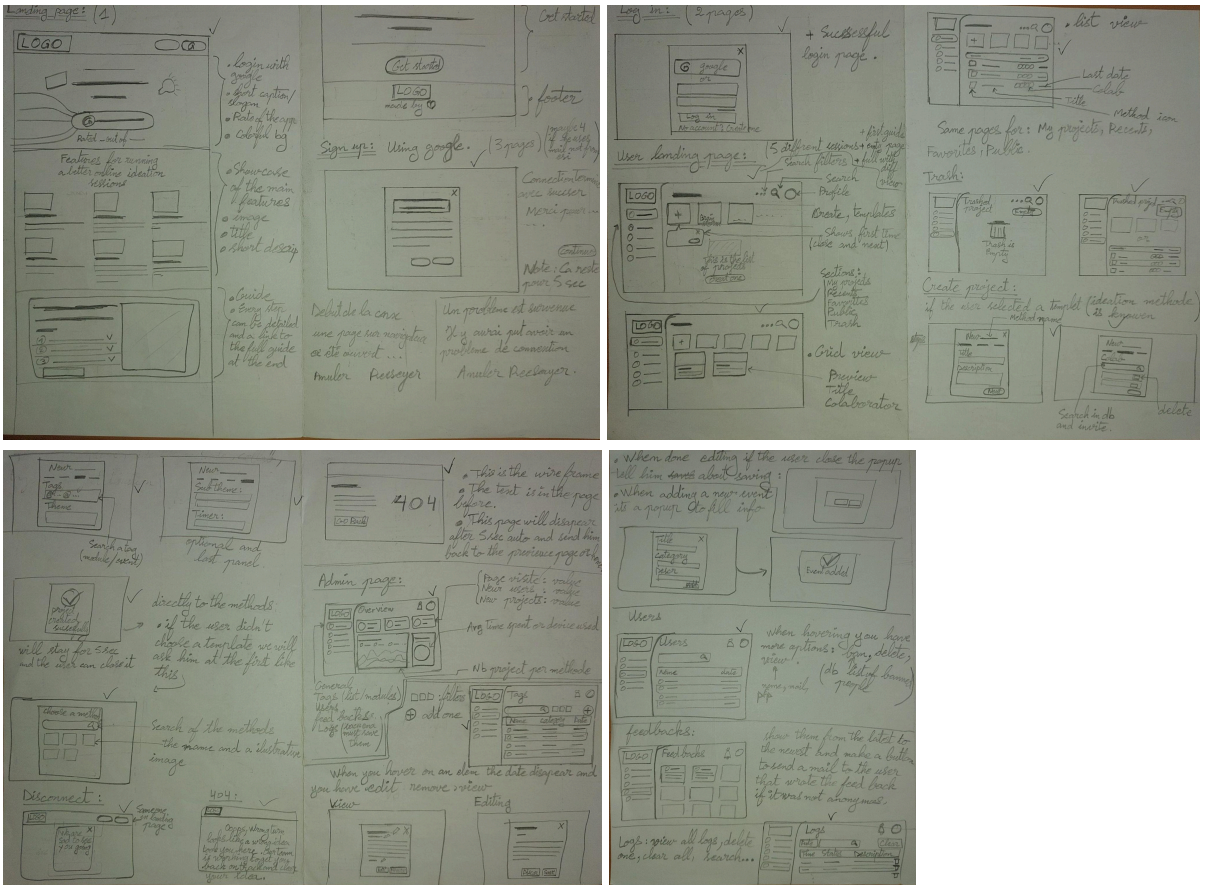
Le violet : créativité, originalité, et mystère.

Le vert : Fraîcheur, croissance, et nature.

G. Interface utilisateur

1. Wireframe

Avant d'entamer le travail concret de conception et de développement, il est essentiel de visualiser les futures interfaces de l'application. Pour cela, nous avons créé à la main des wireframes, des représentations visuelles simplifiées et schématiques des différentes pages et fonctionnalités de l'application.

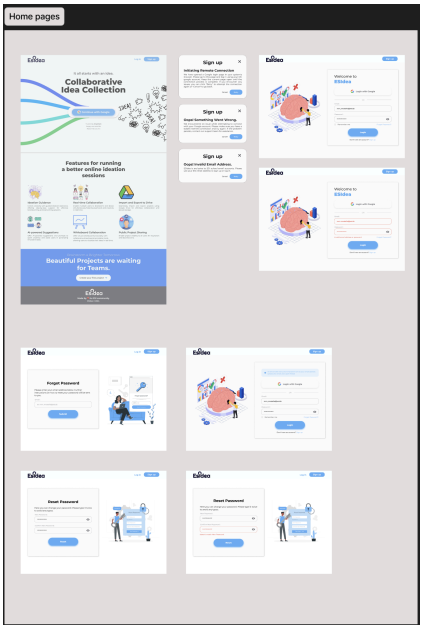


2. Maquette de l'interface utilisateur

Dans ce qui suit, nous montrerons les principales pages de notre application, réalisées lors de la phase de conception en utilisant le logiciel Figma - logiciel de design UI/UX - en suivant les croquis réalisés (wireframes).

- **Partie visiteur**

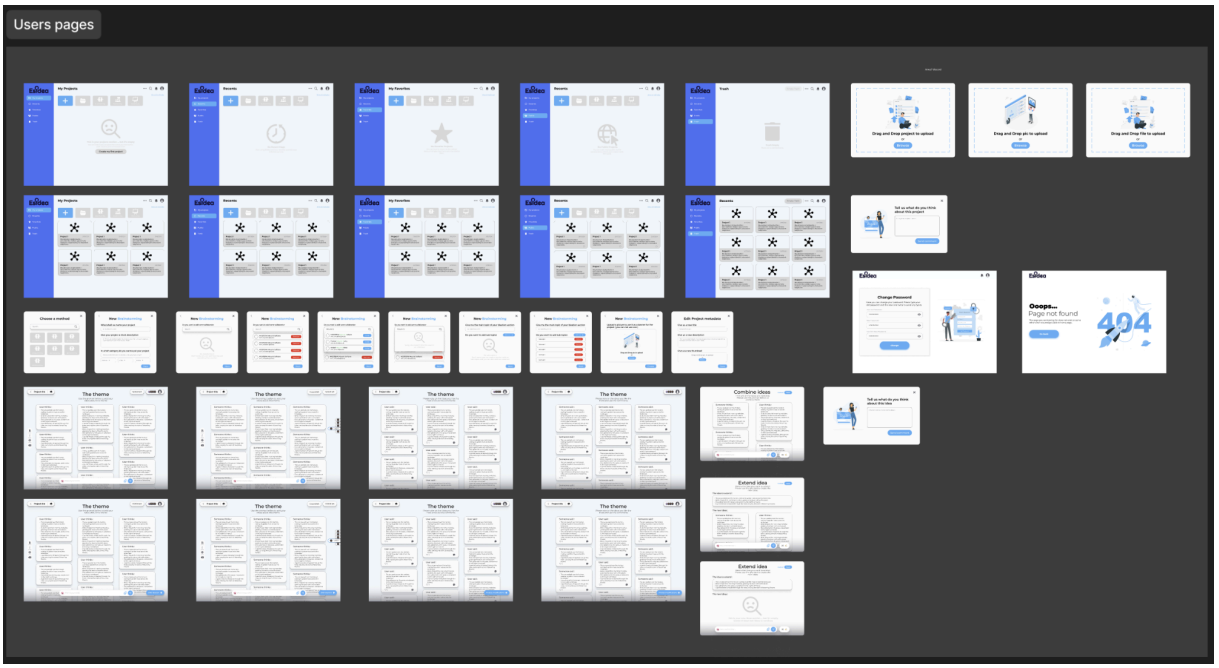
Cette partie est destinée aux visiteurs de notre application avant de se connecter, que ce soit en créant un nouveau compte, en utilisant leur compte existant, ou même en se connectant avec leur compte Google.

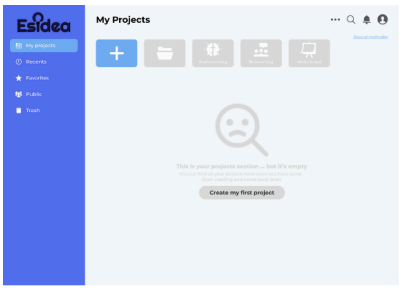


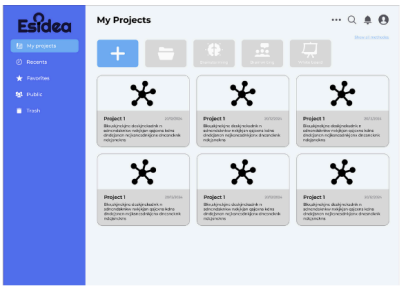
- **Partie utilisateur**

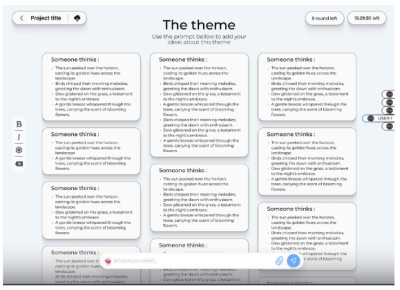
Après la connexion, l'utilisateur fera face à un tableau de bord où il trouvera tous ses projets. Il aura la possibilité de les modifier, supprimer, déplacer et les ouvrir. Il pourra aussi créer de nouveaux projets en utilisant les différentes méthodes d'idéation disponibles.

Chaque projet suivra différentes étapes d'idéation jusqu'à sa finalisation où l'utilisateur sera guidé tout au long du chemin et il sera capable de le visualiser et de l'exporter.



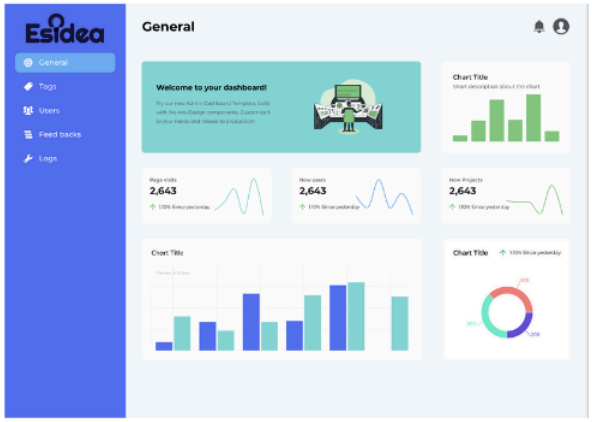


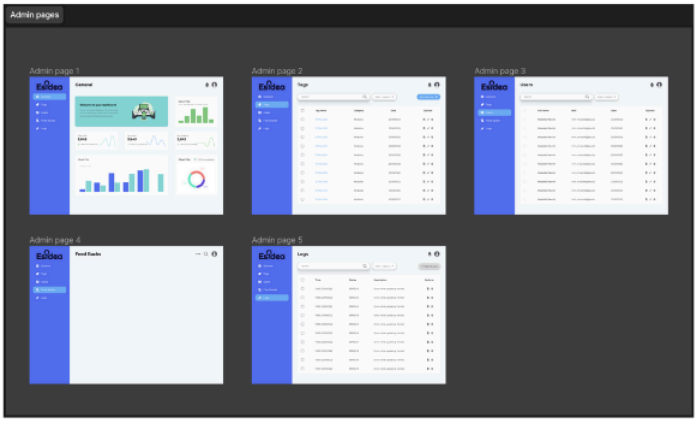




- **Partie administrative**

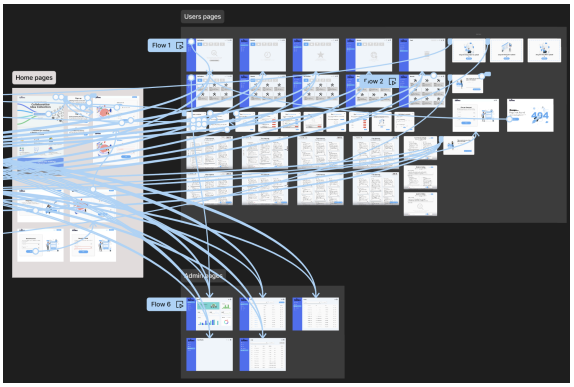
Cette partie concerne l'administrateur de l'application qui aura accès à toutes les statistiques et pourra modifier et gérer les différents paramètres et listes sur l'application.

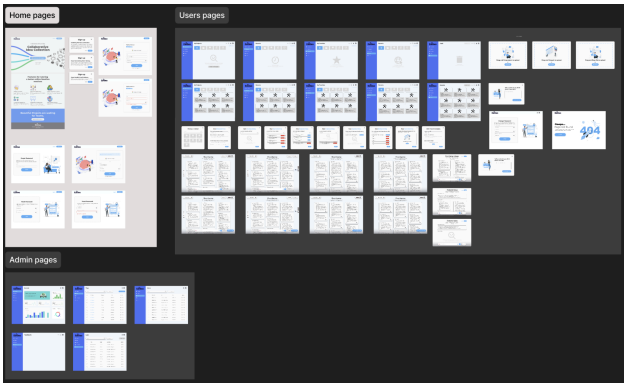




- **Partie Prototypage**

Afin de visualiser les relations entre les pages, Figma nous donne la possibilité de créer un prototype d'application.





H. Conclusion

C'est ainsi que la partie conception du projet a été réalisée pendant près de trois semaines. Grâce à une bonne documentation, les outils adéquats et un esprit critique et créatif, nous sommes venus à bout de cette phase et prêts à entamer la phase de réalisation.

VI. Réalisation

A. Introduction

Après une phase de conception réussie, il est temps d'entamer la réalisation du logiciel. Cette étape implique le développement distinct des différentes fonctionnalités, à la fois au niveau du front-end et du back-end, suivies de leur intégration pour obtenir un produit final cohérent. Une attention particulière devra être accordée à la réalisation de tests afin d'optimiser et de peaufiner le logiciel, garantissant ainsi sa qualité et sa performance.

B. Outils et technologies utilisés

VSCode: :Un éditeur de code source léger et extensible, il est utilisé pour écrire du code dans différents langages de programmation.

Postman: Postman est un outil de développement API qui permet de tester, de déboguer et de documenter les API.

Git: est un système de contrôle de version permettant de suivre les modifications du code source et de collaborer efficacement sur des projets de développement logiciel.

HTML et CSS: HTML est le langage utilisé pour structurer le contenu des pages web, tandis que CSS permet de styliser et de mettre en forme ce contenu en spécifiant les propriétés visuelles comme les couleurs, les polices et les marges.

Tailwind: est un framework CSS utilisé pour simplifier le développement frontend en fournissant des classes prédéfinies pour styliser rapidement et efficacement les éléments HTML.

JavaScript: JavaScript rend les pages web interactives en ajoutant des fonctionnalités dynamiques comme des animations, des validations de formulaire et des interactions utilisateur avancées.

ReactJS: est une bibliothèque JavaScript populaire pour les interfaces utilisateur interactives et réactives. Avec une approche déclarative et une architecture basée sur les composants, il offre une construction modulaire et une maintenance aisée. Son utilisation d'un DOM virtuel optimise les performances et assure une expérience utilisateur fluide. De plus,son écosystème étendu, incluant des outils comme Redux et React Router, offre une grande richesse fonctionnelle pour la création d'applications web dynamiques et performantes.

TypeScript: TypeScript enrichit JavaScript en introduisant des types statiques, renforçant la fiabilité des applications côté serveur, notamment avec Node.js. Cette fonctionnalité permet de détecter les erreurs précocement, facilitant la documentation du code et améliorant sa qualité et sa maintenabilité. De plus, TypeScript offre beaucoup d'outils, incluant l'achèvement de code et la vérification d'erreurs, ce qui accroît la productivité des développeurs et réduit les temps de débogage. Il prend également en charge les fonctionnalités modernes de JavaScript, comme ES6/ES7, async/await et les décorateurs.

Express: est un framework web pour Node.js, facilitant la création d'applications web en fournissant des fonctionnalités essentielles pour le routage, la gestion des requêtes et des réponses, et la configuration du serveur. Il se distingue par sa rapidité et son minimalisme, offrant une efficacité accrue dans le développement d'applications web.

De plus, Express.js offre un support robuste des middleware, facilitant l'intégration d'éléments tels que l'authentification, la journalisation et la gestion des erreurs. Il simplifie également la gestion des routes, permettant une définition claire pour différents schémas d'URL et méthodes HTTP. Grâce à son architecture modulaire et sa flexibilité, Express.js est hautement scalable, adapté aux petites et même aux grandes applications web.

Node: Node.js, environnement d'exécution JavaScript côté serveur, assure une gestion efficace des opérations d'E/S, favorisant des interactions fluides en temps réel. Sa nature asynchrone et non bloquante lui confère une grande scalabilité, idéale pour le projet Esidea.

Mongodb et Mongoose: MongoDB, une base de données NoSQL orientée documents, s'avère être un choix idéal pour le projet Esidea grâce à sa flexibilité et ses capacités en temps réel. En associant MongoDB à Mongoose, une bibliothèque JavaScript, le projet bénéficie d'une interaction simplifiée et d'une validation efficace des données. MongoDB excelle dans la gestion des mises à jour en temps réel, garantissant que les modifications sont immédiatement reflétées dans l'application, favorisant ainsi des expériences de collaboration transparentes. Son modèle de données flexible permet de stocker et de gérer des données non structurées ou semi-structurées en toute simplicité. De plus, la scalabilité horizontale de MongoDB assure des performances élevées même sous des charges lourdes, contribuant ainsi à une expérience utilisateur fluide en temps réel.

C. Modules secondaires et scripts

Pendant la phase de réalisation, nous avons dû automatiser quelques tâches telles que l'initialisation de la base de données, le chatbot et le déploiement de Docker.

1. Initialisation de la base de données

Pour bien démarrer la réalisation, nous avons commencé par créer un script qui génère des adresses e-mail avec des noms et prénoms algériens, ainsi qu'un autre script en Python qui parcourt les pages web de l'ESI (https://talents.esi.dz/scolar/programme_list/ et https://www.esi.dz/clubs-de-lesi/) pour récupérer les listes des clubs, événements et modules pédagogiques, puis les stocker en base de données, comme le montrent les photos suivantes.


```
{
    "image": "https://www.esi.dz/wp-content/uploads/2021/05/GDG-1-150x150.png",
    "Club": "Google Developer Groups",
    "Date de création": "Septembre 2011",
    "Président(e)": "Redha AMIROUCHE",
    "Domaine": "Développement des produits technologiques",
    "Nombre d'événements": "51",
    "Evénements majeurs": "DevFest"
  },
```

```
{
    "Code": "AL",
    "Precision": "SIQ / SIL",
    "Titre": "Architectures Logicielles",
    "DDC(Domaine de connaissances)": "Ingénierie du Logiciel",
    "Credit": "4",
    "Coef": "4",
    "Edition": "2012",
    "Volume horaire cours": "30.00",
    "Volume horaire TD": "30.00",
    "Volume horaire TP": "–"
  },
```


2. Chatbot

Suite à l'analyse du cahier des charges, nous avons remarqué une fonctionnalité optionnelle que nous avons intégrée dans notre solution : le chatbot. Nous avons créé un script en Python utilisant l'API de Gemini pour permettre des interactions en chat, combiner des idées et même étendre ou simplifier une idée en plusieurs parties.

```
Menu:
1. Talk to AI
2. Combine Ideas
3. Split an Idea
4. Exit
Enter your choice (1-4):
```

3. Lancement de Docker

Afin de simplifier le déploiement du conteneur Docker de notre application et d'aider les utilisateurs peu familiers avec la ligne de commande, nous avons créé un script Bash qui exécute toutes les commandes nécessaires pour démarrer notre site web.

D. Le front-end

1. Introduction

La partie front-end d'une application web est la facette avec laquelle les utilisateurs interagissent directement. Elle englobe l'ensemble des éléments visibles et interactifs de l'interface utilisateur, ainsi que les fonctionnalités qui lui sont associées. La conception et le développement front-end jouent un rôle crucial dans la création d'une expérience utilisateur convaincante, intuitive et réactive.

Dans le développement de la partie front-end de notre application web, nous avons opté pour l'utilisation de React JS et Tailwind CSS.

Pour la réalisation de cette partie cruciale de notre projet, nous avons fait appel à des technologies modernes qui nous ont permis de créer une interface utilisateur dynamique, réactive et esthétiquement plaisante. Parmi ces technologies, deux se distinguent particulièrement: React JS et Tailwind CSS

2. Implémentation de Frontend selon la conception

1/Traduction des maquettes en composants React:

Nous avons commencé par convertir les maquettes de l'interface utilisateur en composants React. Chaque élément visuel a été décomposé en composants réutilisables, en accordant une attention particulière à la modularité et à la réactivité.

2/Utilisation intensive de Tailwind CSS:

Grâce à l'approche "utility first" de Tailwind CSS, nous avons pu traduire facilement les conceptions visuelles en code. En utilisant des classes utilitaires pour définir les styles, nous avons reproduit fidèlement les maquettes tout en bénéficiant d'une personnalisation rapide et d'une base de code légère.

3/Intégration de la navigation et des interactions utilisateur:

Nous avons implémenté la navigation entre les différentes pages de l'application ainsi que les interactions utilisateur telles que les clics, les survols et les saisies. Cela a été réalisé en utilisant les fonctionnalités de routage de React et en intégrant des événements interactifs aux composants.

4/Tests et optimisation continus:

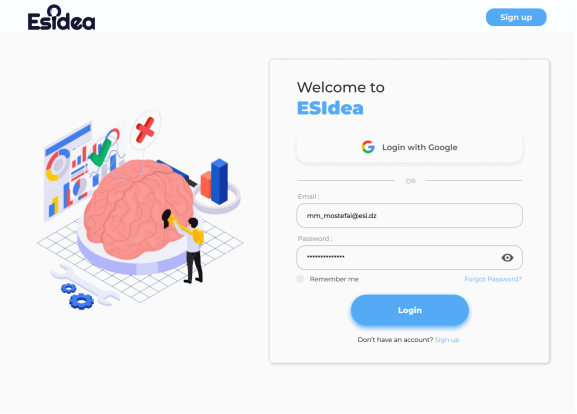
Nous avons mené des tests approfondis pour garantir la stabilité et la performance de l'interface utilisateur. Cela comprenait des tests d'accessibilité, des tests de compatibilité entre navigateurs et des tests de charge pour identifier et résoudre les éventuels problèmes.

En suivant cette approche, nous avons réussi à concrétiser notre vision conceptuelle en une interface utilisateur fonctionnelle et esthétiquement cohérente. Chaque décision prise lors de

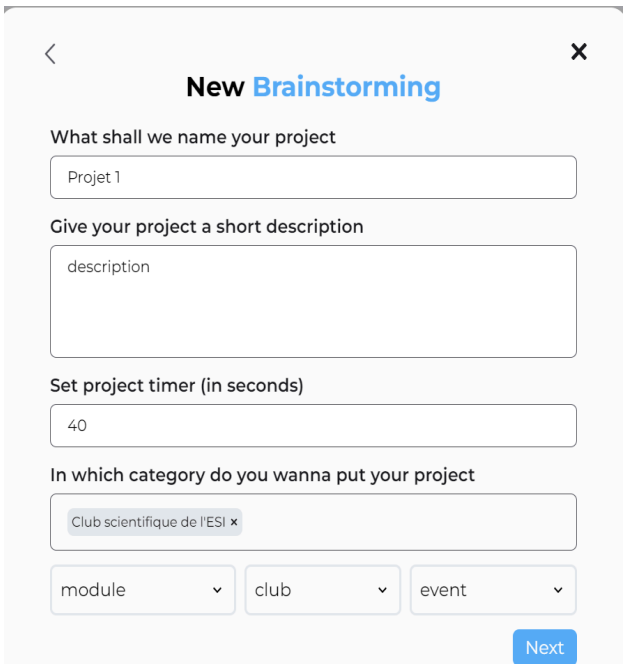
la phase de conception a été soigneusement traduite en code frontend, aboutissant à une expérience utilisateur engageante.

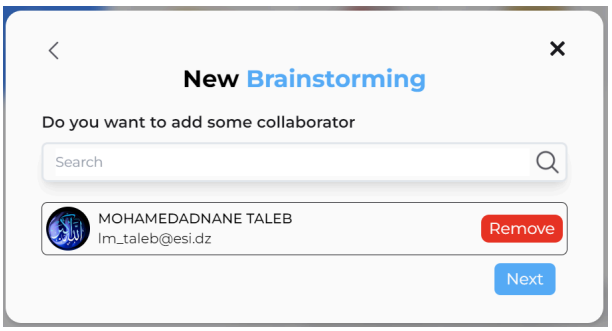
3. Présentation du Front-end

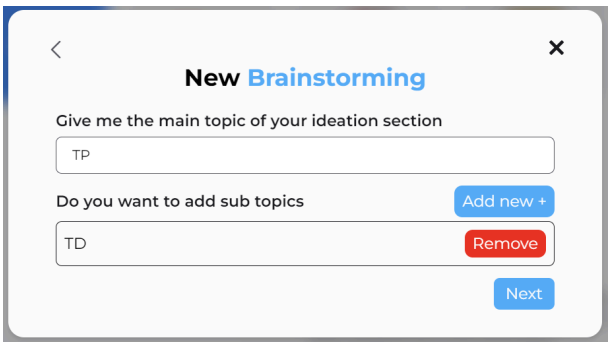
1. Page d'accueil et authentification:

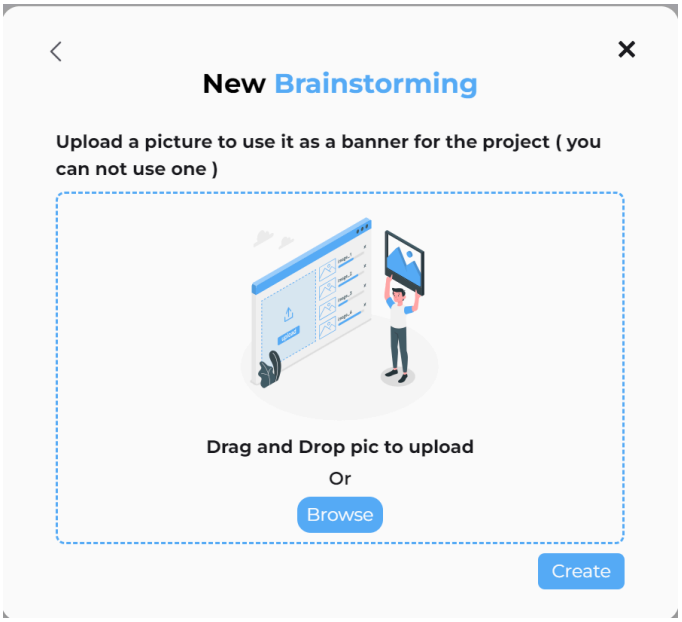


2. Création du projet:

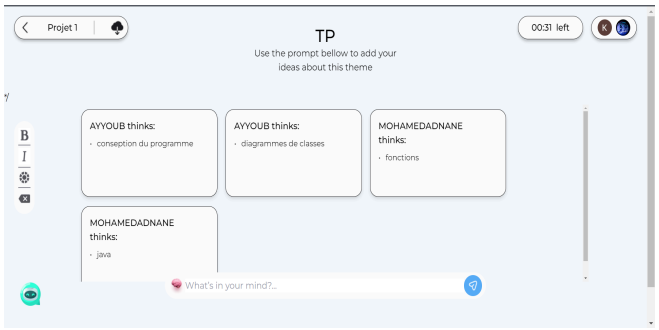




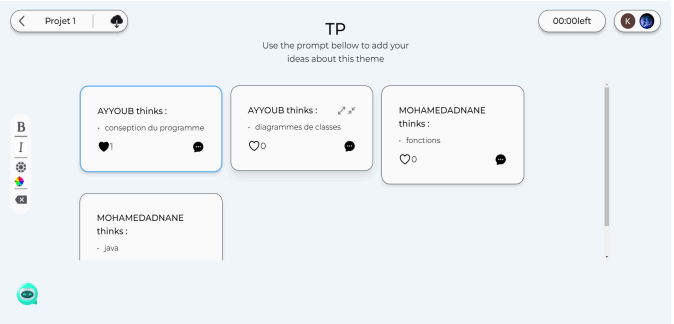




3. Page d'idéation en brainstorming:

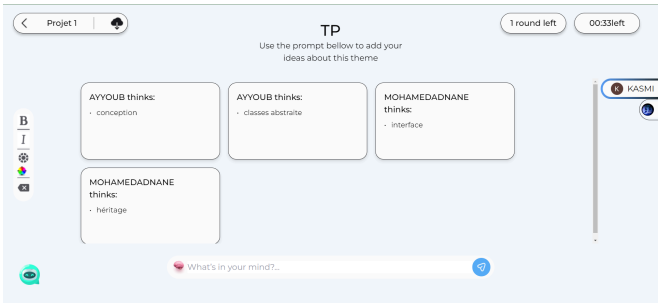


Mode écriture

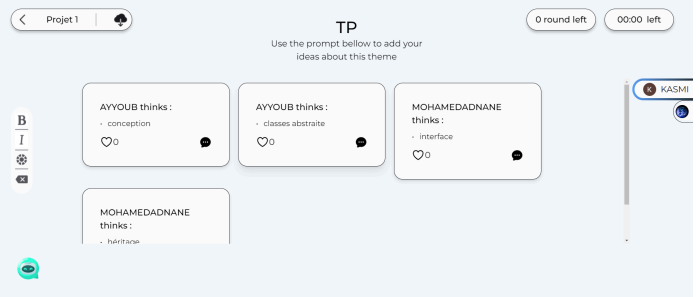


Mode évaluation

4. Page d'idéation en brainwriting:

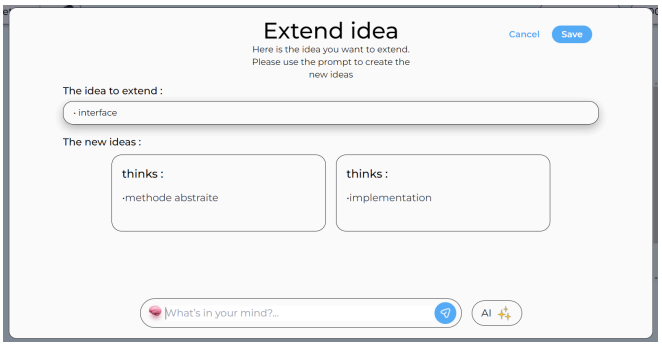


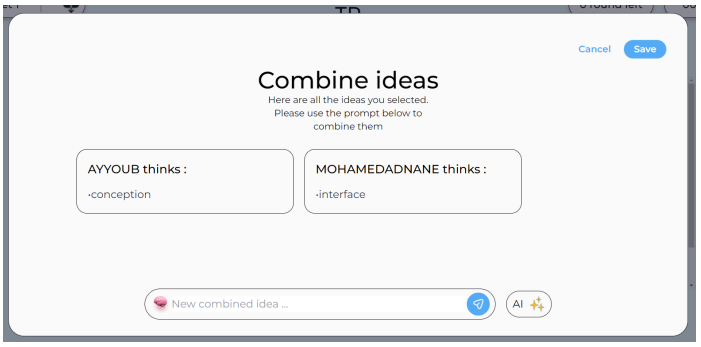
Mode écriture



Mode évaluation

5.Manipulation:

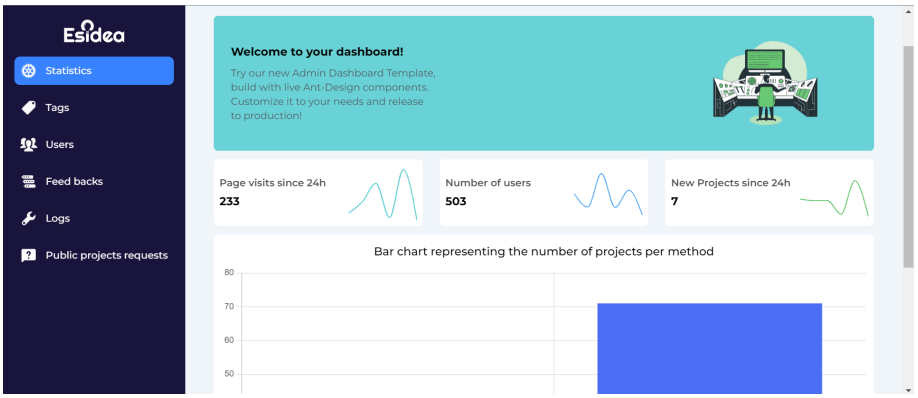




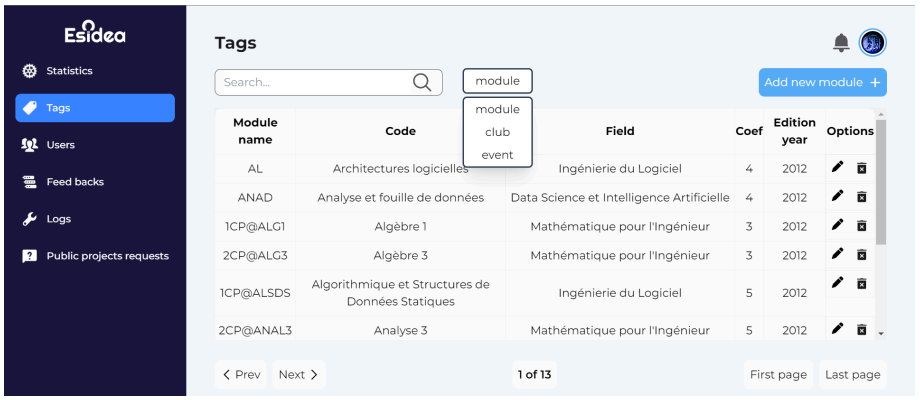
Expansion

Combination

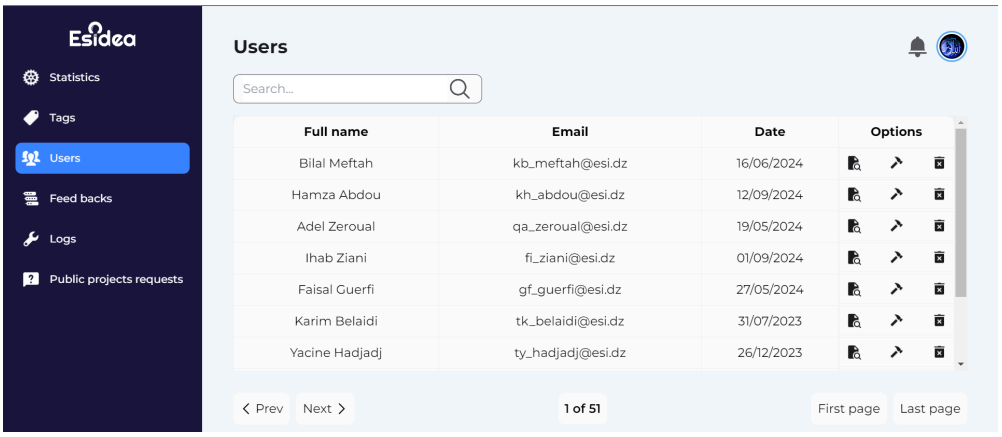
6. Page de statistiques:



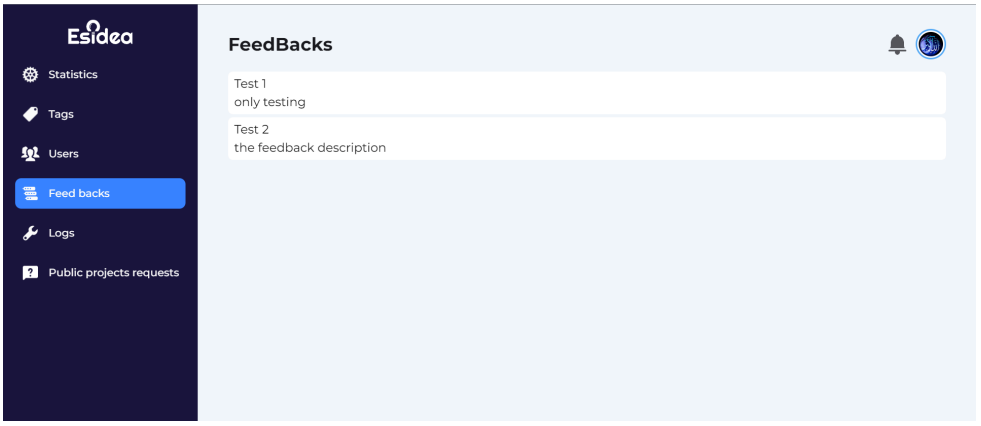
7. Page de traitement des listes:



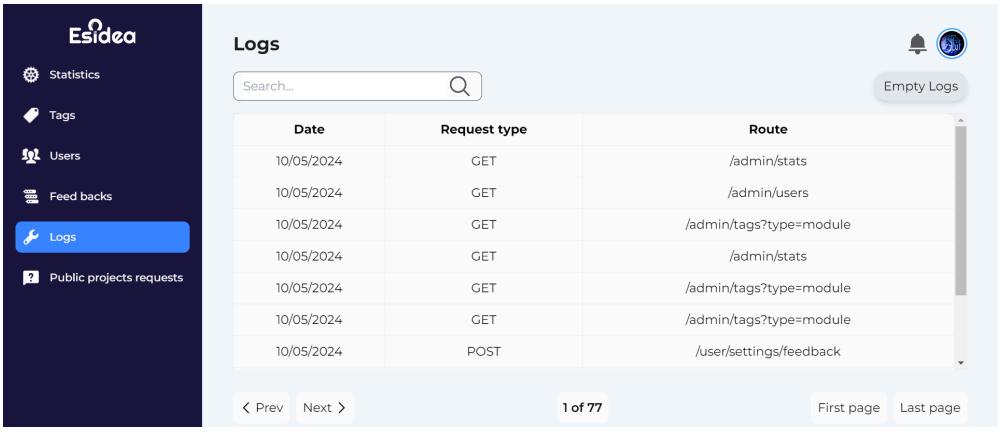
8. Page d’utilisateurs:



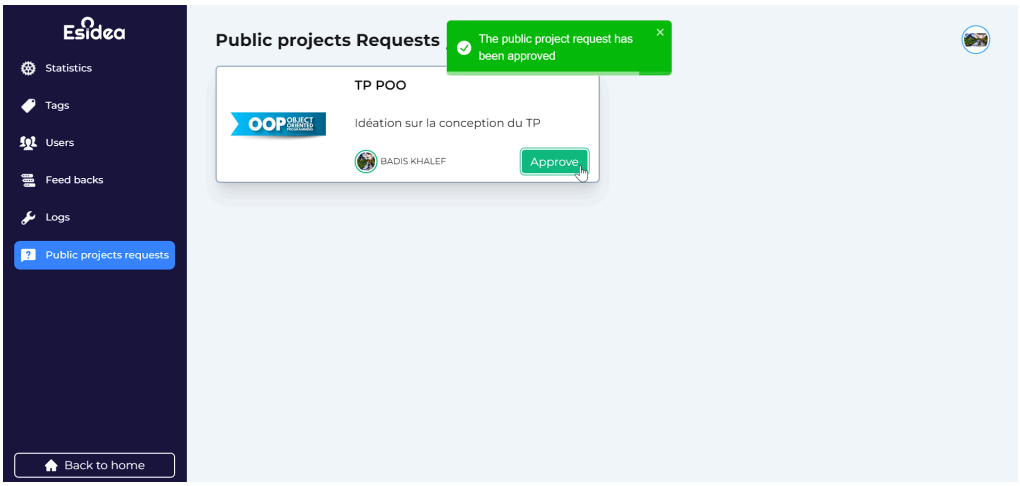
9, Page d'avis:



10, Page des journaux administratifs:



11. Page de demandes de publication des projets



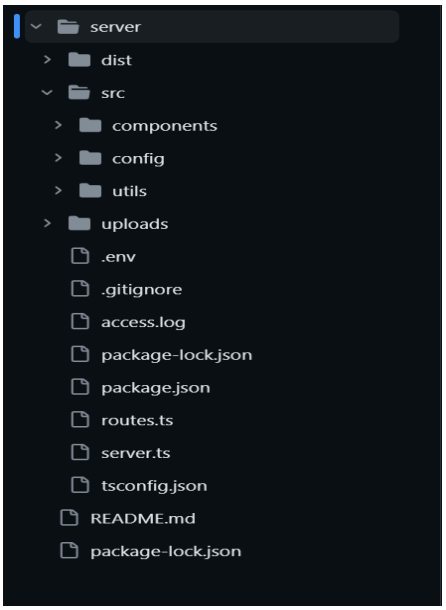
E. Le back-end

1. Introduction

Le backend de l'application web est responsable de la gestion des opérations côté serveur qui soutiennent la fonctionnalité et les performances de l'application. Il gère des tâches telles que le traitement des requêtes provenant du frontend, l'exécution de la Business logic, l'interaction avec les bases de données, et la gestion des processus d'authentification et d'autorisation. En essence, le backend agit comme le moteur de l'application, orchestrant le flux de données et assurant le bon fonctionnement de tout le système.

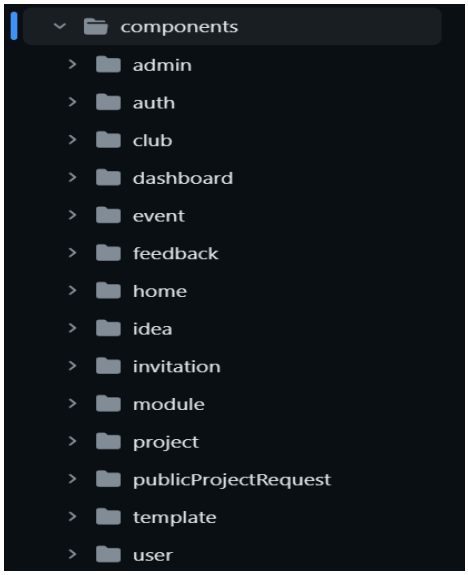
Le backend du site web Esidea est développé avec TypeScript et Node.js, en utilisant le framework Express. MongoDB est utilisé comme base de données principale. Cette pile technologique facilite le développement fluide et la gestion efficace du traitement des données, de la logique métier et des points d'API, garantissant des performances optimales et une fiabilité accrue.

2. Architecture et structure de code



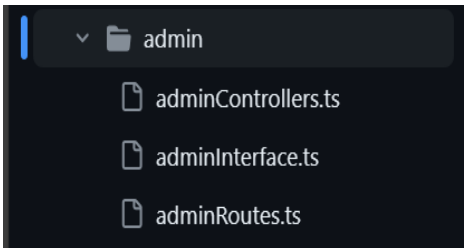
L'architecture backend du site web Esidea suit une architecture monolithique. Dans cette structure, tous les composants et fonctionnalités sont contenus dans une seule base de code. La structure du code est organisée en différents répertoires dans le dossier `/server/src` :

- `/components` : Contient des sous-répertoires pour différents composants de l'application, tels que admin, user, project, etc. Chaque composant a son propre ensemble de contrôleurs, routes, modèles et interfaces.

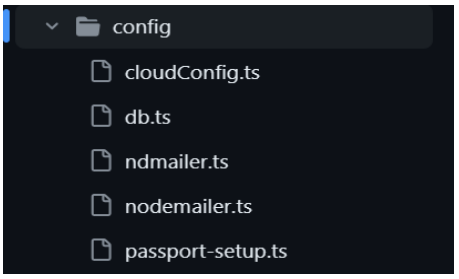


`/admin` : Contient des contrôleurs, routes, modèles et interfaces spécifiques à la fonctionnalité admin.

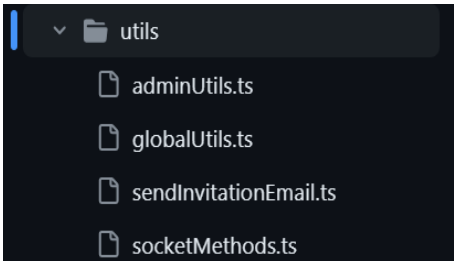
- `/user`, `/project`, etc. : Répertoires similaires pour d'autres composants.



- `/config` : Contient des fichiers de configuration pour la configuration du serveur, les connexions de base de données et d'autres paramètres spécifiques à l'environnement.



- `/utils` : Contient des fonctions utilitaires pouvant être utilisées dans différentes parties de l'application.



Le fichier `server.ts` sert de point d'entrée pour l'application backend, où le serveur est initialisé et configuré. Le fichier `routes.ts` définit les routes pour différents points d'API et les associe aux fonctions de contrôleur correspondantes.

Les fichiers `package.json` et `package-lock.json` contiennent des métadonnées sur les dépendances et configurations du projet.

Implémentation des APIs

Structure des points d'accès : Les points d'accès sont organisés en composants logiques tels que user, project et admin, chacun avec son propre ensemble de routes pour gérer des fonctionnalités spécifiques.

Principes RESTful : Les points d'accès de l'API respectent les principes RESTful, en utilisant les méthodes HTTP standard pour les opérations CRUD (par exemple, GET, POST, PUT, DELETE) sur des ressources telles que les utilisateurs, les projets et les données admin.

Nomination des ressources : Les noms des points d'accès sont choisis de manière à être descriptifs et indicatifs de la ressource avec laquelle ils interagissent. Par exemple, les routes pour les opérations liées à l'utilisateur se trouvent sous le point d'accès /user, tandis que celles pour les tâches liées à l'admin se trouvent sous /admin.

Formats de requête et de réponse : Les requêtes et les réponses suivent des formats normalisés, généralement des charges utiles JSON. Les requêtes incluent les données nécessaires à l'opération, et les réponses fournissent des informations pertinentes ou des confirmations.

Gestion des erreurs : Les points d'accès gèrent les erreurs en renvoyant des codes d'état HTTP appropriés accompagnés de messages d'erreur informatifs. Cela aide les applications clientes à comprendre la nature de l'erreur et à prendre des mesures appropriées.

Authentification et autorisation : Les points d'accès nécessitant une authentification ou une autorisation sont correctement sécurisés à l'aide de mécanismes tels que les jetons JWT ou

l'authentification basée sur des sessions. L'accès à certaines routes peut être restreint en fonction des rôles ou des permissions des utilisateurs.

Validation et assainissement : Les données d'entrée transmises aux points d'accès sont validées et assainies pour prévenir les vulnérabilités de sécurité. Cela inclut la validation des paramètres de requête, la garantie de l'intégrité des données et l'assainissement des entrées utilisateur pour prévenir les attaques par injection.

exemple d'un API : supprimer un utilisateur

```
const deleteUser = async (req: Request, res: Response) => {
  const errResult = validationResult(req)
  if (!errResult.isEmpty())
    return res.status(400).send({ errors: errResult.array() })

  const userId: string = req.body.id

  if (!isObjectIdOrHexString(userId)) {
    return res
      .status(400)
      .send({ error: 'Bad id must be 24 character hex string' })
  }
  const objectId = new mongoose.Types.ObjectId(userId)

  try {
    const user = await User.findById(objectId)
    if (!user) {
      return res.status(404).send({ error: 'User not found' })
    }
    for (const projectObj of user.projects) {
      const project = await Project.findById(new mongoose.Types.ObjectId(projectObj.project._id.toString()))
      if (project) {
        const updatedCollaborators = project.collaborators.filter(
          collaborator => collaborator.member._id.toString() !== userId
        )
        project.collaborators = updatedCollaborators
        await project.save()
      } else {
        return res.status(404).send({ error: 'Project of the user not found' })
      }
    }
    await User.deleteOne(objectId)
    return res.sendStatus(200)
  } catch (error) {
    console.log(error)
    return res.sendStatus(500)
  }
}
```

le code ci-dessus illustre ces points :

Validation de la requête : La fonction valide les données de la requête entrante pour s'assurer qu'elles respectent le format et les critères requis. Si la validation échoue, une réponse 400 Bad Request est renvoyée au client, avec des détails sur les erreurs de validation.

Suppression d'utilisateur : Après validation réussie, la fonction procède à la suppression de l'utilisateur du système. Elle envoie une réponse 200 OK pour indiquer que l'opération a réussi.

Gestion des erreurs : Toutes les erreurs rencontrées lors de la validation de la requête ou de la suppression de l'utilisateur sont traitées de manière appropriée. Une réponse générique 500 Internal Server Error est renvoyée au client en cas d'erreur inattendue.

3. Implémentation de la base de données

Dans l'application web Esidea, l'implémentation de la base de données tourne autour de MongoDB, une base de données NoSQL offrant flexibilité et évolutivité pour stocker différents types de données de manière efficace. Le backend utilise Mongoose, une bibliothèque ODM (Object Data Modeling) pour MongoDB et Node.js, pour définir des schémas et interagir avec la base de données.

Les schémas sont définis à l'aide de Mongoose pour représenter la structure des documents stockés dans les collections MongoDB. Par exemple, le schéma User définit des champs tels que prénom, nom, email et mot de passe, ainsi que leurs types de données respectifs et les règles de validation associées. De manière similaire, des schémas sont définis pour d'autres entités telles que les projets, les idées, les sujets et les collaborations, reflétant leurs attributs spécifiques et leurs relations.

```
const ideaSchema = new Schema<IdeaInterface>({
  createdBy: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'User',
    required: true
  },
  projectId: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Project',
    required: true
  },
  content: {
    type: String,
    required: true
  },
  creationDate: {
    type: Date,
    default: Date.now()
  },
  topic: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Topic',
    required: true
  },
  comments: [
    {
      content: String,
      author: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'User'
      }
    }
  ],
  votes: {
    type: Number,
    default: 0
  },
  isBold: Boolean,
  isItalic: Boolean,
  color: String
})
```

Les interfaces sont utilisées pour imposer une vérification de type et fournir un contrat pour les formes de données. Dans le projet ESIDEA, des interfaces TypeScript sont employées pour définir la structure des documents de schéma. Par exemple, l'interface User spécifie les champs et leurs types de données, garantissant la cohérence entre la définition du schéma et les données réellement stockées dans la base de données.

```
export interface IdeaInterface extends Document {
  createdBy: UserInterface
  projectId: ProjectInterface
  content: string
  creationDate: Date
  topic: TopicInterface
  comments: {
    content: string
    author: UserInterface
  }[]
  votes: number
  isBold: boolean
  isItalic: boolean
  color: string
}
```

En utilisant des schémas et des interfaces, le backend garantit l'intégrité des données, facilite la maintenance du code et permet une interaction transparente avec la base de données

MongoDB. De plus, le typage statique de TypeScript améliore la productivité des développeurs en détectant les erreurs potentielles lors de la compilation, réduisant ainsi la probabilité de problèmes lors de l'exécution. Globalement, l'implémentation de la base de données dans le projet ESIDEA respecte les meilleures pratiques pour une gestion efficace des données et le développement robuste de l'application.

4. Authentification et autorisation

1. Authentification

Authentification Google : Les utilisateurs s'authentifient en utilisant Google OAuth, garantissant que seuls les membres de l'ESI peuvent accéder au site web. Cela implique de rediriger les utilisateurs vers le service d'authentification de Google, où ils se connectent avec leurs identifiants ESI.

Authentification par Email et Mot de passe : Les utilisateurs peuvent également se connecter en utilisant leur email et leur mot de passe s'ils ont préalablement créé un compte. Cela implique de vérifier leurs identifiants par rapport aux données stockées dans la base de données.

2. Autorisation

Distinction entre Admin et Utilisateur : Après l'authentification, le backend distingue entre les administrateurs et les utilisateurs réguliers en fonction de leurs rôles. Cette distinction détermine le niveau d'accès et les permissions accordées à chaque utilisateur.

Contrôle d'accès : Les mécanismes d'autorisation contrôlent l'accès à différentes parties de l'application en fonction des rôles des utilisateurs. Les administrateurs ont des privilèges élevés, ce qui leur permet d'effectuer des tâches administratives et d'accéder à des fonctionnalités restreintes.

Middleware : Les fonctions middleware interceptent les requêtes entrantes pour vérifier l'identité et les permissions de l'utilisateur avant de lui accorder l'accès aux ressources protégées. Cela garantit que seuls les utilisateurs autorisés peuvent effectuer certaines actions ou accéder à des points d'accès spécifiques.

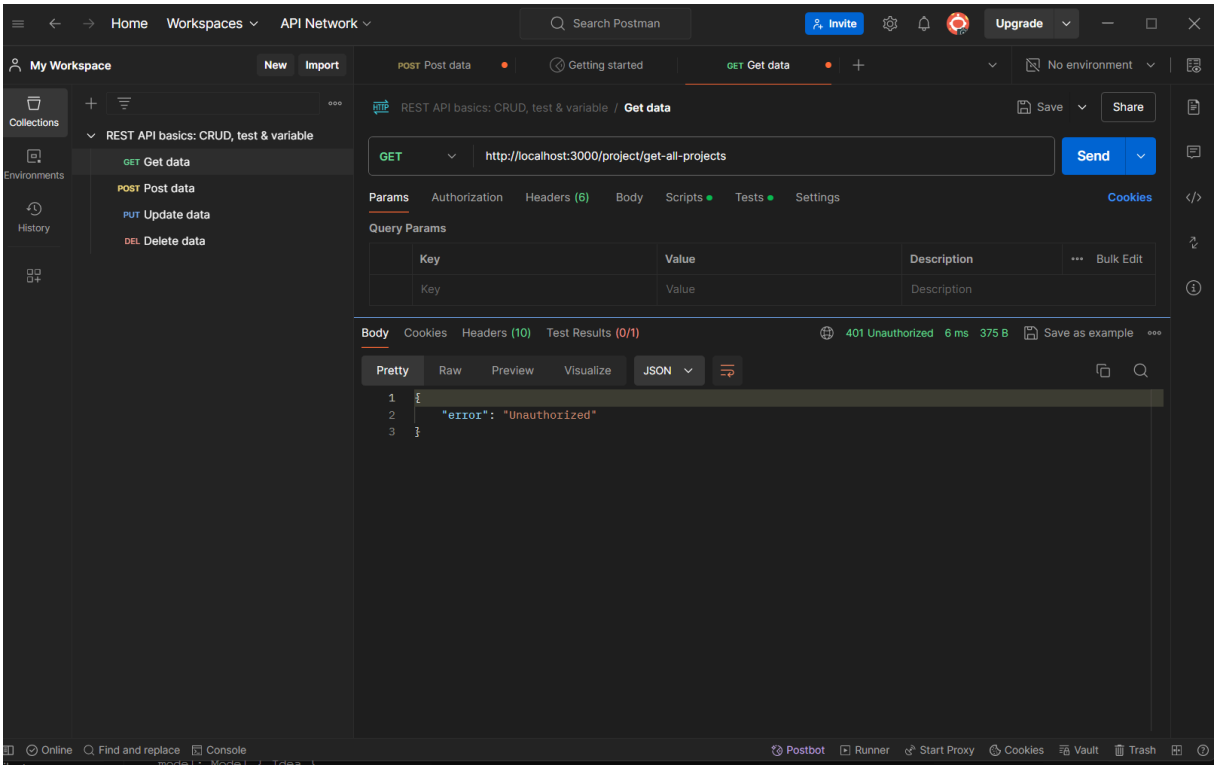
L'exemple ci-contre illustre un middleware qui vérifie si l'utilisateur est authentifié. Celui-ci est appliqué à toutes les API nécessitant l'identification de l'utilisateur

```
export const authMiddleWare = async (
  req: Request,
  res: Response,
  next: NextFunction
) => {
  const authHeader = req.headers.authorization
  if (!authHeader || !authHeader.startsWith('Bearer')) {
    return res.status(401).json({
      error: 'Unauthorized'
    })
  }
  const token: string = authHeader.split(' ')[1]
  try {
    const payload: AuthPayload = jwt.verify(
      token,
      process.env.JWT_SECRET as string
    ) as AuthPayload

    const userId = payload.userId
    const email = payload.email
    req.user = { userId, email }
    next()
  } catch (err) {
    console.log(err)
    return res.status(401).json({
      error: 'Unauthorized'
    })
  }
}
```

5. Testes

Dans le cadre du projet ESIDEA, l'outil Postman a été employé pour procéder à des tests objectifs des API backend. Par le biais de Postman, des requêtes ont été méticuleusement acheminées vers les points d'accès du serveur, et les réponses qui en découlent ont été minutieusement analysées afin de déterminer dans quelle mesure les fonctionnalités du backend se conforment aux spécifications préétablies. Postman a permis de valider différents scénarios, couvrant notamment l'authentification des utilisateurs, la récupération et la manipulation des données, grâce à la configuration précise des requêtes avec des paramètres et des charges utiles adéquats. L'usage de Postman a garanti une évaluation rigoureuse de la justesse et de la fiabilité des API backend avant leur déploiement.



6. Conclusion

En conclusion, l'infrastructure backend du projet ESIDEA, développée avec rigueur et expertise, tire parti de technologies modernes telles que TypeScript, Node.js, Express.js et MongoDB. Cette combinaison permet d'assurer une gestion efficace des données, une scalabilité optimale et une fiabilité accrue des fonctionnalités. Les tests rigoureux effectués avec des outils comme Postman garantissent la qualité et la fiabilité des API backend. Ainsi, le projet est bien positionné pour offrir une application de haute qualité et évolutive à ses utilisateurs.

F. L'Intégration

l'intégration désigne le processus structuré de fusion de divers éléments, notamment le frontend et le backend, les interfaces de programmation d'applications externes (API), les systèmes de gestion de bases de données, entre autres. Cette fusion vise à créer une application opérationnelle et cohérente. Elle requiert généralement l'établissement de connexions entre ces composants afin de permettre une communication transparente et une interaction efficace. Cette intégration favorise ainsi une expérience utilisateur fluide et assure le bon fonctionnement de l'application. Par ailleurs, elle peut englober la gestion des interactions entre les différents modules, la synchronisation des données ainsi que la garantie de la sécurité et des performances générales de l'application.

1. Intégration Frontend-Backend

Esidea a opté pour une intégration harmonieuse du frontend, élaboré avec la technologie React.js, avec le backend conçu sur la base de Node.js. Cette démarche vise à faciliter la communication entre les divers composants de l'application, tant du côté client que du côté serveur. Pour ce faire, Axios, un client HTTP largement reconnu pour sa polyvalence aussi bien dans les navigateurs que dans l'environnement Node.js, est employé afin de réaliser des requêtes HTTP asynchrones entre le frontend et le backend. De plus, afin de répondre aux exigences de collaboration en temps réel, l'intégration de fonctionnalités telles que Socket.io a été privilégiée, permettant ainsi une synchronisation efficace des données et des interactions en temps réel entre les différents utilisateurs de la plateforme.

2. Intégration de l'Authentification Google

Esidea intègre une fonctionnalité de chatbot pour améliorer l'engagement des utilisateurs et fournir de l'assistance. Le chatbot est intégré en tant qu'API, permettant une communication transparente entre le frontend et le service de chatbot. Les utilisateurs peuvent interagir avec le chatbot via l'interface frontend, et le backend gère la communication avec le service de chatbot pour traiter les requêtes des utilisateurs et fournir des réponses pertinentes.

3. Intégration de l'API de Chatbot

Esidea intègre une fonctionnalité de chatbot pour améliorer l'engagement des utilisateurs et fournir de l'assistance. Le chatbot est intégré en tant que API, permettant une communication transparente entre le frontend et le service de chatbot. Les utilisateurs peuvent interagir avec le chatbot via l'interface frontend, et le backend gère la communication avec le service de chatbot pour traiter les requêtes des utilisateurs et fournir des réponses pertinentes.

G. Documentation du logiciel

La documentation est un élément indispensable lors du développement d'un logiciel, son but est de fournir toute l'aide et les informations nécessaires sur l'utilisation, le fonctionnement et la maintenance de l'application. Les utilisateurs se réfèrent toujours à la documentation en cas de problème lors de leur utilisation.

Dans le cadre de notre projet, et selon les demandes de notre CDC, il nous est demandé d'établir trois documents pour la documentation de l'app. Une notice d'installation, une notice d'utilisation ainsi qu'une aide en ligne disponible via l'application.

1. Manuel d'installation

Ce manuel ou guide fournit les instructions étape par étape sur la configuration nécessaire pour pouvoir utiliser notre site web. Il a pour but de faciliter cette tâche à tout utilisateur, qu'il soit ou non familier avec ce type de logiciel. Il donne aussi la possibilité à l'utilisateur non familier avec Docker de lancer automatiquement le site web à travers le script d'auto-lancement.

2. Manuel d'utilisation

Malgré que l'interface de ESIdea est intuitive, on a créé un guide d'utilisation qui a pour but de présenter à l'utilisateur toutes les fonctionnalités de l'application, comment l'utiliser et comment effectuer certaines actions. A travers des instructions claires et des captures d'écran jointes, ce guide est assez simple à suivre et compréhensif pour un utilisateur lambda.

3. Aide en ligne

L'aide en ligne est accessible via le site en cliquant sur le profil et puis "Help" . Elle détaille toutes les fonctionnalités proposées par l'application et tous les renseignements liés sous forme de questions et réponses. Elle facilite la recherche de solutions aux problèmes rencontrés par les utilisateurs.

H. Conclusion

C'est ainsi que la phase de réalisation prend fin, cette étape a été une expérience enrichissante et gratifiante. Grâce au travail d'équipe, à l'utilisation des outils adéquats et notre engagement constant, nous sommes venus à bout du projet et nous sommes parvenus à réaliser une application répondant aux besoins du cahier des charges.

Nous restons conscients qu'il y a toujours place à l'amélioration et à l'ajout de nouvelles fonctionnalités. Nous continuerons à travailler sur notre application, en tenant compte des retours des utilisateurs, tout en cherchant à perfectionner l'expérience utilisateur.

VII. Conclusion

Notre projet s'est articulé autour de la problématique consistant à concevoir un logiciel d'idéation destiné aux membres de notre école, en tirant parti de diverses techniques telles que le brainstorming et le brainwriting pour faciliter le processus d'idéation. Nous avons débuté par une analyse approfondie des applications similaires, en identifiant leurs limitations et en déterminant les fonctionnalités spécifiques à intégrer dans l'environnement de l'ESI.

L'étape suivante a impliqué une analyse détaillée du cahier des charges, ainsi que l'élaboration d'un planning minutieux et la prise en compte proactive des risques potentiels. Ensuite, nous avons procédé à la conception détaillée et à la mise en place des outils de communication et de collaboration, tels que Discord et Git, afin de favoriser une collaboration efficace entre les membres de l'équipe.

Le développement du UI/UX et des composants back-end et front-end a ensuite été entrepris, suivi de l'intégration, une phase cruciale pour assurer la cohérence et la fluidité de l'ensemble du système. Les tests et corrections ont été réalisés de manière continue tout au long du projet, avec une attention particulière accordée à cette étape lors de la phase finale. La dockerisation a été mise en œuvre pour permettre le partage et la remise de notre application aux encadrants.

Enfin, une documentation exhaustive du logiciel, comprenant des manuels et une aide en ligne, a été élaborée avec soin pour faciliter l'utilisation de l'application et accompagner les utilisateurs dans leur parcours. Cette documentation a été complétée par la rédaction du présent rapport, offrant une rétrospective détaillée de notre travail et de nos réalisations au cours de ce projet.

A. Compétences acquises

1. **Gestion de projet** : Nous avons pu gérer les différentes phases du projet en planifiant, suivant et anticipant les risques potentiels, ce qui a contribué à respecter les délais et à maintenir la cohésion de l'équipe.
2. **Collaboration et communication** : Nous avons travaillé en équipe en partageant nos idées et en résolvant les défis ensemble, ce qui a favorisé un environnement de travail productif.
3. **Résolution de problèmes** : Malgré les défis techniques rencontrés, nous avons pu identifier des solutions et les mettre en pratique de manière systématique, permettant ainsi de surmonter tous les obstacles rencontrés dans le cadre de notre projet.
4. **Programmation** : Malgré notre manque d'expérience préalable, nous avons développé des compétences en développement web pour créer des fonctionnalités complexes.
5. **Tests et débogage** : Nous avons effectué des tests réguliers pour identifier et corriger les erreurs, garantissant ainsi la qualité et la fiabilité de l'application.
6. **Documentation** : Nous avons élaboré des manuels d'utilisation détaillés pour accompagner les utilisateurs dans l'utilisation de l'application, assurant ainsi une expérience utilisateur fluide.
7. **Apprentissage autodidacte** : Nous avons développé des compétences d'auto-apprentissage en explorant de nouvelles technologies et en consultant des ressources en ligne pour acquérir les compétences nécessaires au projet.

8. **Adaptabilité** : Notre capacité à nous adapter aux changements et à trouver des solutions face aux défis imprévus a contribué dans notre succès.

B. Perspectives

Perspectives de fonctionnalités envisagées mais non finalisées :

1. **Exportation et importation de projets :** Intégrer la possibilité de sauvegarder les projets en externe et d'importer d'autres depuis diverses sources, y compris les services de stockage cloud.
2. **Événements personnalisés :** Offrir aux utilisateurs la capacité d'ajouter des événements personnalisés dans leur propre liste, offrant ainsi une expérience plus personnalisée et adaptée à leurs besoins individuels.
3. **Support multilingue :** Élargir la portée du site en proposant des versions dans plusieurs langues, telles que le français, l'arabe, etc., pour une accessibilité accrue et une meilleure expérience utilisateur globale.
4. **Personnalisation de l'interface :** Permettre aux utilisateurs de personnaliser l'apparence de l'interface en choisissant entre des thèmes clairs ou foncés, ainsi que des palettes de couleurs personnalisées, afin de répondre à leurs préférences esthétiques.

Perspectives du panneau administrateur :

1. **Statistiques améliorées :** Intégrer des statistiques plus détaillées, telles que la moyenne de temps passé sur le logiciel, pour fournir à l'administrateur une vision plus approfondie de l'utilisation de la plateforme.
2. **Fonctionnalités de gestion des utilisateurs :** Compléter la fonctionnalité de blocage d'utilisateurs pour permettre à l'administrateur de suspendre temporairement les comptes en cas de besoin, et développer le module de feedback pour permettre à l'administrateur de répondre aux commentaires des utilisateurs.

Perspectives à long terme :

1. **Amélioration de la fonctionnalité de projet public :** Élargir la fonctionnalité de projet public pour créer un véritable réseau social dédié à l'Esi, où les membres peuvent partager, évaluer et discuter des projets, favorisant ainsi une culture de collaboration et d'innovation au sein de la communauté.
2. **Contrôle d'accès basé sur les rôles :** Envisager la mise en place d'un système de contrôle d'accès avancé basé sur les rôles, différenciant les utilisateurs étudiants et enseignants. Cette fonctionnalité viserait à fournir des expériences personnalisées en fonction du rôle de l'utilisateur, permettant aux enseignants d'accéder aux outils administratifs et de superviser les activités des étudiants. Pendant ce temps, les étudiants bénéficient de l'accès à des outils de collaboration sur les projets et à des ressources éducatives.