



RAPPORT DU STAGE DE PERFECTIONEMENT

Conception et développement d'une solution de suivi des stagiaires

Élaboré par : **Bouzayani Malek**

Encadré par : **Ms Mejai Rabii**

Organisme d'accueil : **NEOLEDGE**



Période : du au

Année Universitaire : 2020/2021

Validation finale

Par :

Signature:.....

Sommaire

Remerciement	6
Introduction Générale.....	7
Chapitre1	8
1) Introduction.....	8
2) Présentation de l'entreprise	8
3) Présentation du Projet	8
4) Analyse de l'existant	9
5) Les objectifs	9
6) Conclusion	9
Chapitre2.....	10
1) Introduction.....	10
2) Identification des acteurs.....	10
3) Identification des besoins fonctionnels.....	10
4) Identification des besoins non-fonctionnels	11
5) Conception	11
a. Diagramme de cas d'utilisation.....	11
b. Diagramme de classe	12
6) Conclusion	13
Chapitre3.....	14
1) Introduction.....	14
2) Environnement logiciel	14
3) Technologies Utilisé	15

a. Back-End.....	15
b. Front-End.....	16
4) Développement du back-end	16
5) Développement des Différents Interfaces du Front-end.....	19
6) Conclusion	27
Conclusion	28
Bibliographie	29

Liste des figures

Figure 1: Logo Néobelge	8
Figure 2: Diagramme des cas d'utilisation.....	12
Figure 3: Diagramme des classes	13
Figure 4 : Visual Studio Code.....	14
Figure 5: Microsoft Visual Studio.....	14
Figure 6: Postman.....	15
Figure 7 : .NetCore Framework.....	15
Figure 8 : VueJS	16
Figure 9: Model de donnée	16
Figure 10: Base de données.....	17
Figure 11: Diagrammes de la base de données	17
Figure 12: Liaison avec la base de données	18
Figure 13: Requêtes http.....	18
Figure 14: Test et correction avec Postman	19
Figure 15: Interface du l'authentification	19
Figure 16: Interface d'accueil.....	20
Figure 17: Interface édit utilisateur	20
Figure 18: Interface ajoute utilisateur	21
Figure 19: Interface supprimer utilisateur	21
Figure 20: Interface liste des stages	22
Figure 21 : Interface ajoute stage.....	22
Figure 22: Chercher stage.....	23
Figure 23: Interface édit stage	23

Figure 24 : Interface affecté utilisateur	24
Figure 25: Interface publié stage.....	24
Figure 26: Interface liste stagiaire	25
Figure 27: Interface détail stagiaire.....	25
Figure 28: Interface ajouter commentaire	26
Figure 29: Interface accepté stagiaire.....	26
Figure 30 : Interface stagiaire accepter	27

Remerciement

Tout d'abord, Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage, qui m'ont accompagné et aidé lors de la rédaction de ce rapport.

J'adresse mes remerciements au directeur de l'entreprise **Mr Sami Mcharek** qui m'a permis de postuler pour ce stage.

Je tiens à remercier vivement mon encadrant de stage **Mr Rabii Mejai** ainsi que toute l'équipe de **NeoLedge Station** pour l'accueil, le temps passé ensemble et le partage de leurs expertises au quotidien. Grâce aussi à leur confiance, j'ai pu m'accomplir totalement dans mes missions et ainsi développé mes connaissances grâce à leur aide précieuse.

Je tiens à exprimer mes profonds respects et gratitude à la direction de mon institut **ISSET DJERBA** qui m'a offert la chance d'effectuer ce stage afin de développer mes compétences et de découvrir de plus en plus l'environnement professionnel.

Introduction Générale

Avec une entreprise de grande taille, recrutant un nombre de stagiaires important chaque année, il serait un peu difficile aux responsables de ressources humaines de suivre tous les stagiaires appliquant à la différente offre de stage.

D'où la nécessité d'une solution pour faciliter cette tâche de ces responsables. **Neledge Station** a pensé de créer une solution de gestion des stagiaires pour résoudre ce problème.

Chapitre1

1) Introduction

Dans ce chapitre, nous intéressons à présenter l'entreprise dans laquelle le stage a été effectué. Ensuite, nous découvrirons le projet à développer, et finalement nous nous intéresserons davantage à la méthodologie adoptée pour réaliser la solution.

2) Présentation de l'entreprise

Depuis bientôt 25 ans, les créateurs de Neolledge, l'activité GED / ECM d'Archimed, n'ont jamais perdu de vue leur valeur fondatrice : l'innovation. Portée par elle au quotidien, elle incite à rechercher sans cesse des nouvelles solutions, pour accompagner leurs clients dans leur transformation digitale et la recherche de performance. Leurs solutions rejoignent cet esprit. Aujourd'hui, ils sont reconnus comme un éditeur de logiciels majeur et à dimension internationale.

Station Neolledge un centre de développement ouvert par Neolledge au sein du Zarzis smartcenter au niveau de parc d'activités économiques de Zarzis depuis janvier 2021 spécialisé dans l'édition de solutions d'ECM.



Figure 1: Logo Néobelge

3) Présentation du Projet

C'est une application web de suivi des stagiaires muni tout d'abord d'une interface qui va permettre aux stagiaires d'appliquer aux stages offertes par l'entreprise en spécifiant les informations demandées.

Ensuite, le responsable de ressources humaines ou de recrutement de l'entreprise aurait la possibilité à travers des différentes interfaces de consulter gérer les offres de stages, gérer la liste de stagiaires, gérer les utilisateurs de l'application ... Dans le but de faciliter ces tâches journalières.

4) Analyse de l'existant

La société NOELEDGE a ouvert une nouvelle succursale à Zarzis spécialisée dans les solutions logicielles Elise et DocFactory. Mais en raison de la création récente de cette succursale et par manque d'expérience, la société reste encore incapable de bien organiser les stagiaires et améliorer les conditions des stages c'est pour cela cette application à lieu.

5) Les objectifs

- ❖ Informatiser le processus de stages concernant la société.
- ❖ Garder l'historique des stages.
- ❖ Faciliter la demande du stage au ceins de la société.
- ❖ Organiser les stages.

6) Conclusion

Durant ce chapitre, nous avons représenté l'entreprise d'accueil et le contexte général du projet. Dans le prochain chapitre, nous discuterons à propos de la planification et les spécifications requises pour la solution.

Chapitre2

1) Introduction

Dans ce chapitre, nous allons extraire les acteurs clé du projet, les différents besoins fonctionnels et non-fonctionnels, les tâches et les parties exécutées au cours du programme, les objectifs du projet ainsi que la présentation des différents diagrammes UML.

2) Identification des acteurs

Ils existent deux acteurs pour cette solution :

- ❖ Les stagiaires qui appliquent pour un stage en remplissant le formulaire développé pour ce but.
- ❖ Le responsable de ressources humaines de l'entreprise qui est le bénéficiaire principal de notre application qui aurait la possibilité de faciliter la tâche de classification et suivie de stagiaires.

3) Identification des besoins fonctionnels

Les besoins fonctionnels représentent les principales fonctionnalités que l'utilisateur attend du logiciel, ils spécifient une fonction qu'un système ou un composant du système doit pouvoir exécuter. Le produit doit donc :

Permet au responsable de ressources humaines :

- ❖ L'authentification.
- ❖ La création d'un stage en ajoutant tous les détails nécessaires.
- ❖ La consultation de la liste des stages ainsi que celle des candidats par stage.
- ❖ L'affectation des différents candidats aux différents stages
- ❖ La consultation des profils de différents stagiaires et le parcours de la liste de tous les stagiaires afin de choisir le plus adapté au poste.
- ❖ L'ajout des commentaires concernant un stagiaire.

Permet au stagiaire :

- ❖ L'application pour un stage en remplissant un formulaire ainsi que le dépôt de son CV.
- ❖ La réception d'un mail de confirmation lors de dépôt de la candidature, ainsi qu'un mail lors de l'acceptation ou le refus.

4) Identification des besoins non-fonctionnels

Les besoins non-fonctionnels sont les exigences qui spécifient les critères qui peuvent être utilisés pour juger du fonctionnement d'une solution, plutôt que des comportements spécifiques. On peut ainsi les distinguer :

- ❖ Offrir des interfaces simples et conviviales pour l'interaction avec l'utilisateur.
- ❖ Contenir un design attirant.

5) Conception

Au cours de cette partie, nous représentons les différents diagrammes UML :

a. Diagramme de cas d'utilisation

Ce premier diagramme va donner une vue générale sur la solution à développer en démontrant les interactions entre les différents acteurs ainsi que les différents cas d'utilisation.

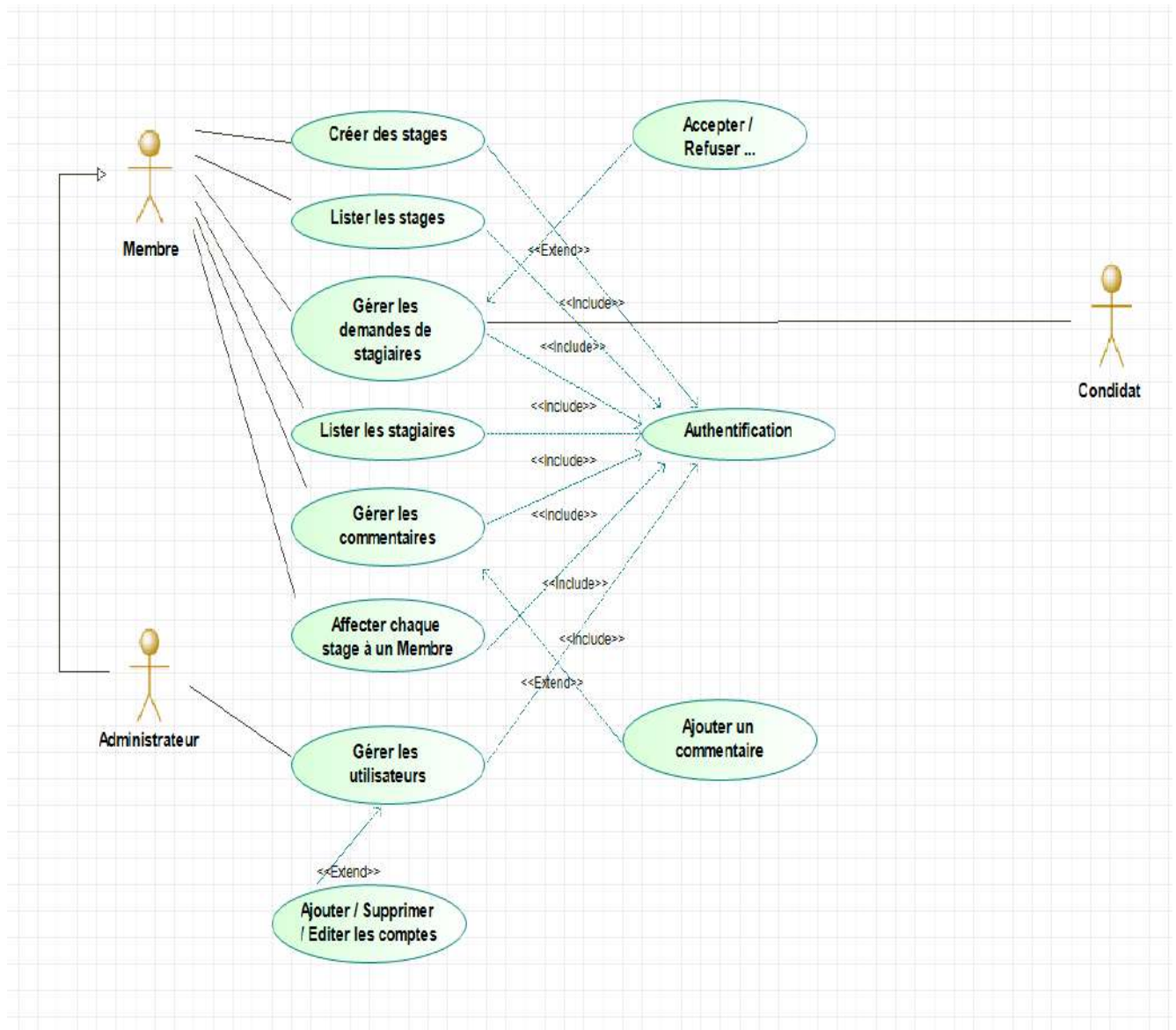


Figure 2: Diagramme des cas d'utilisation

b. Diagramme de classe

À travers ce diagramme nous représentons les différentes entités constitutives de notre solution ainsi que les relations entre eux ainsi que nous donner un Aperçu sur la conception de notre base de données.

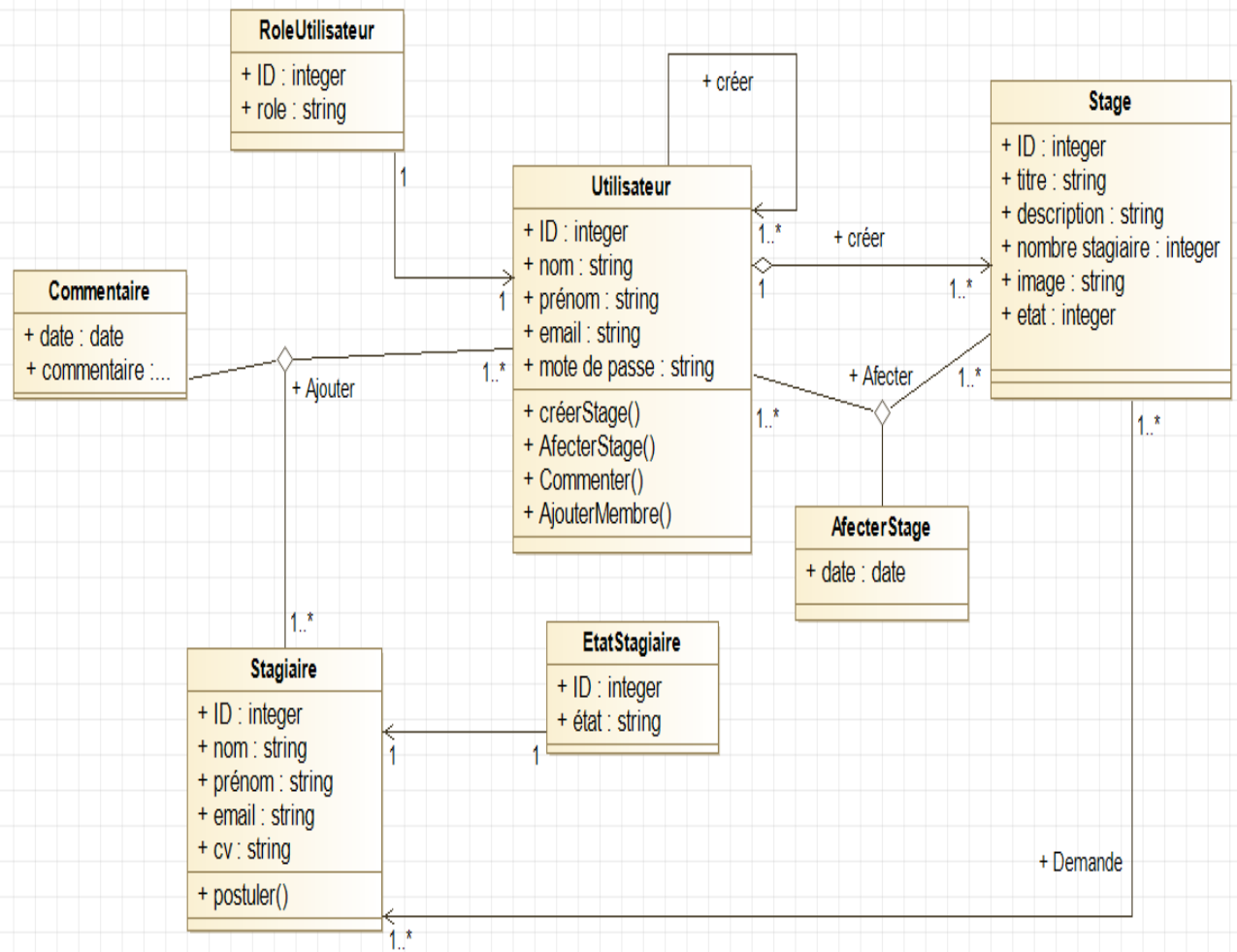


Figure 3: Diagramme des classes

6) Conclusion

Au cours de ce chapitre, nous avons abordé l'analyse des besoins de notre solution afin de découvrir de plus l'apport de notre application pour la société ainsi, nous avons réussi à dégager les principaux acteurs mis en jeu ainsi que la conception des besoins fonctionnels généraux. Dans le prochain chapitre, nous aborderons la réalisation de la solution.

Chapitre3

1) Introduction

Dans ce dernier chapitre, nous présentons les différentes technologies utilisées, les différentes tâches effectuées ainsi que les différents résultats obtenus.

2) Environnement logiciel

Pour le développement de Front-end nous avons utilisé IDE Visual Studio Code un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS.



Figure 4 : Visual Studio Code

Pour le développement du Back-end Nous avons Utilisé l'IDE Microsoft Visual Studio qui est une suite de logiciels de développement pour Windows et mac OS conçue par Microsoft.



Figure 5: Microsoft Visual Studio

Ainsi, nous avons utilisé Postman qui est une plate-forme d'API pour la création et l'utilisation d'API. Postman simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin que vous puissiez créer de meilleures API, plus rapidement.



Figure 6: Postman

3) Technologies Utilisé

a. Back-End

Pour le back-end (service web api) nous avons opté à utiliser Le Framework Libre de Microsoft .NetCore qui est développé pour les systèmes d'exploitation Linux, Microsoft et MacOS. Avec sa première version publiée en 2016, .Netcore est écrite avec des langages de programmation orientées objets tel que le C++ et le C#.



Figure 7 : .NetCore Framework

b. Front-End

Pour la Partie du Front-End nous avons choisir de développer à l'aide du Framework Vues qui est un Framework JavaScript open-source utilisé pour construire des interfaces utilisateur et des applications web mono pages. Ainsi il est considéré comme le Framework le plus performant en comparaison avec les autres Framework développés en JavaScript.



Figure 8 : VueJS

4) Développement du back-end

- Nous Avons Commencé le développement du back-end par la création du projet ainsi que la création de différente model de donnée.

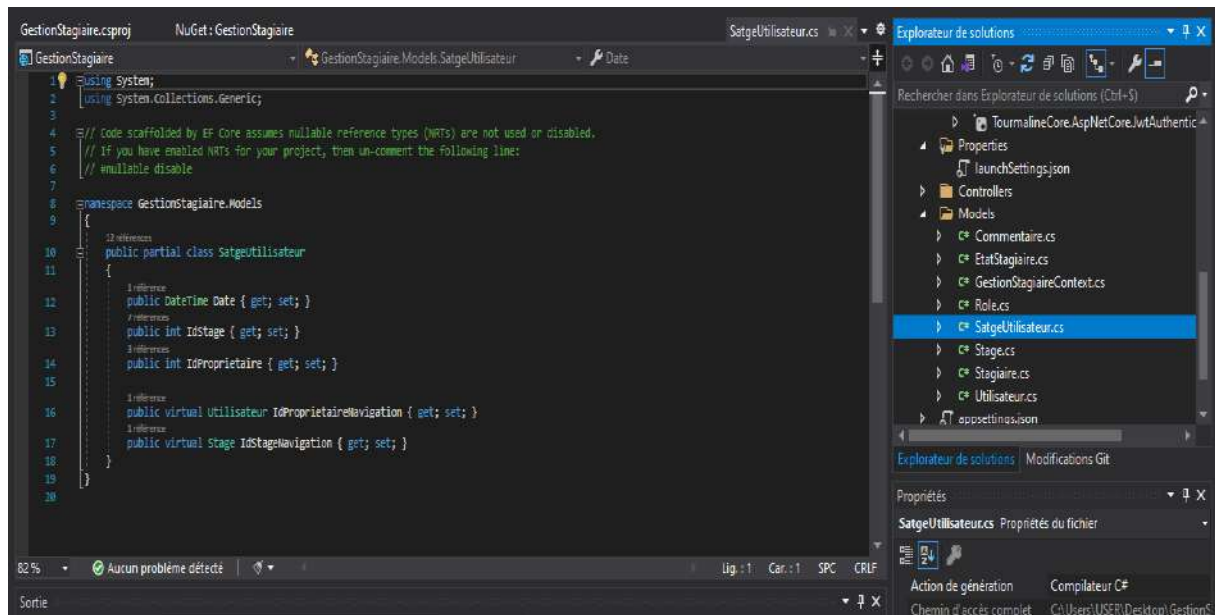


Figure 9: Model de donnée

- Nous avons passé ensuite à la création de notre base de données à l'aide des requêtes SQL en introduisant les différentes relations entre les différentes tables.

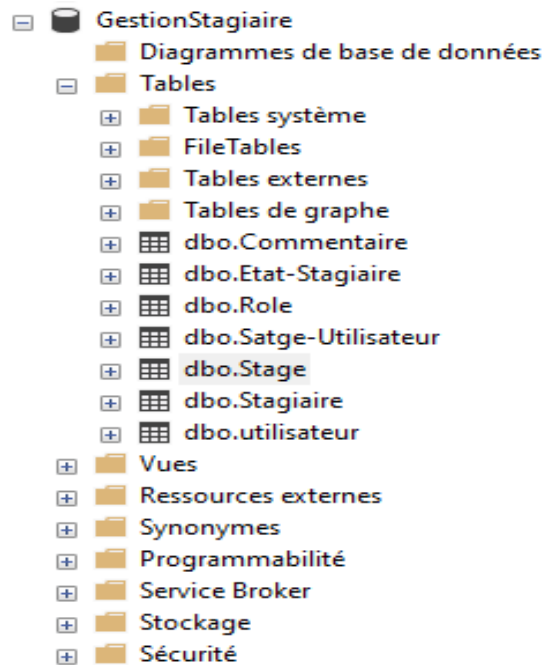


Figure 10: Base de données

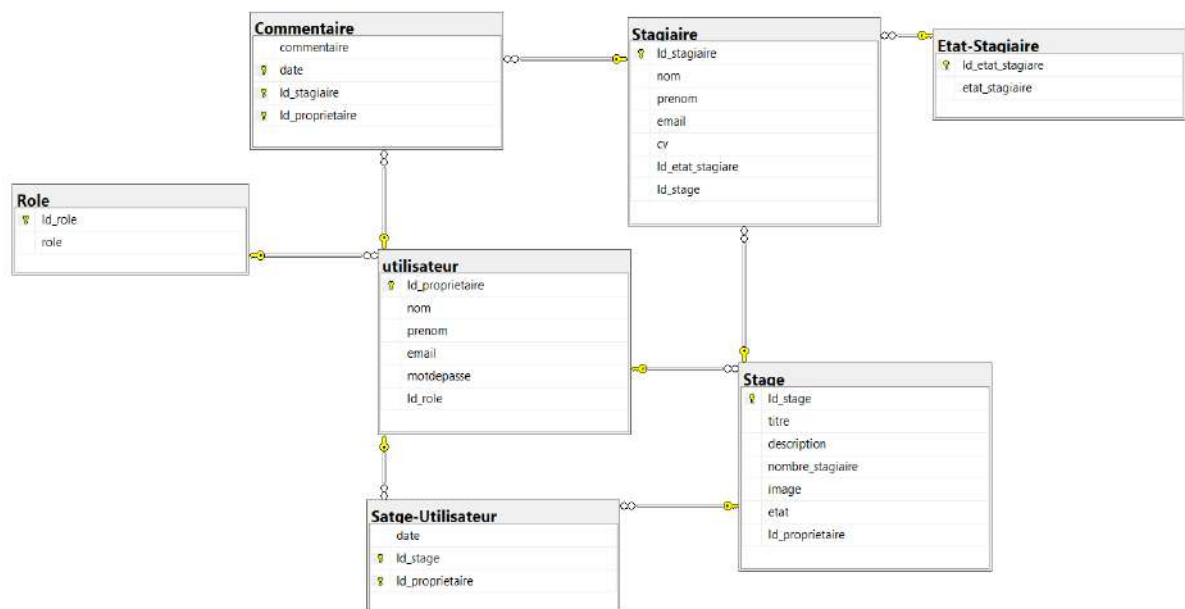
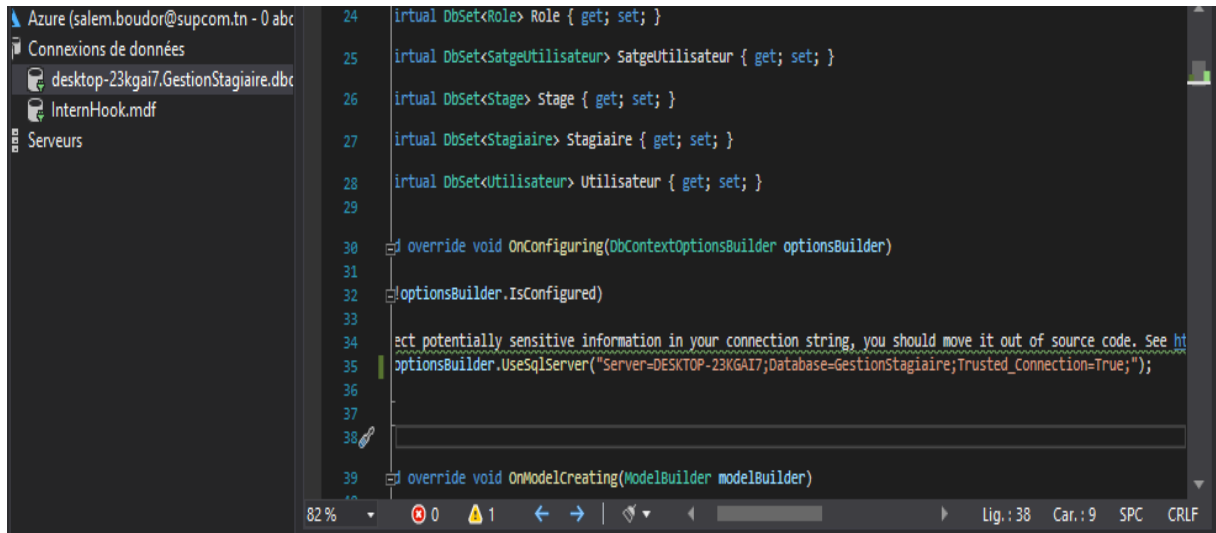


Figure 11: Diagrammes de la base de données

- Nous avons lié notre base de données à notre application où les modèles de données sont déjà créés.



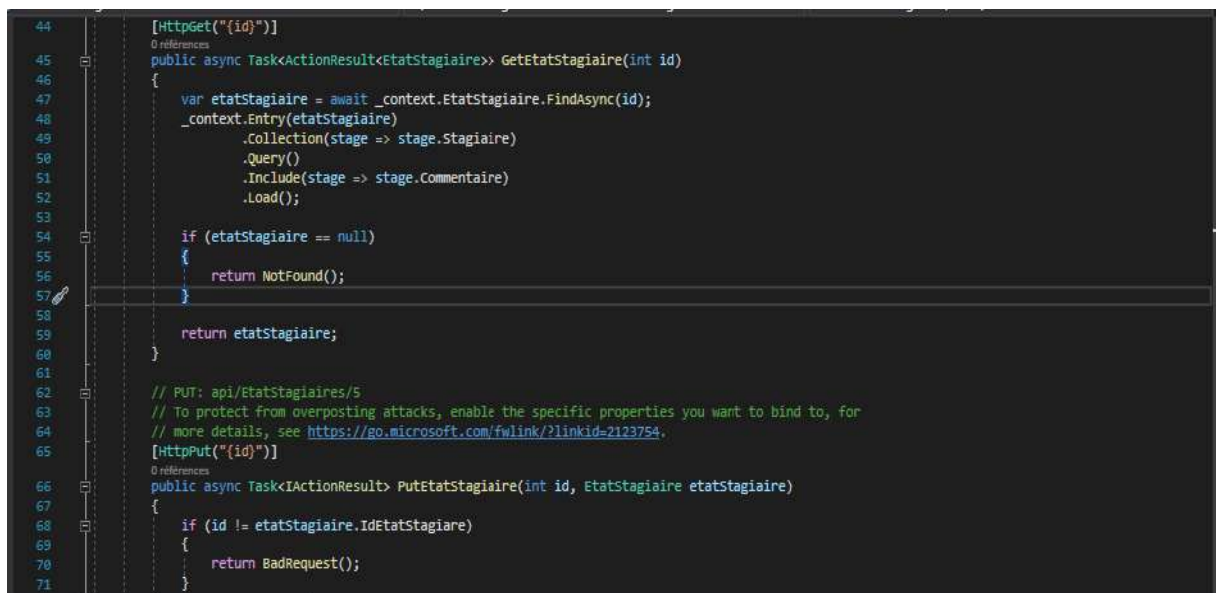
```

24  virtual DbSet<Role> Role { get; set; }
25  virtual DbSet<SatgeUtilisateur> SatgeUtilisateur { get; set; }
26  virtual DbSet<Stage> Stage { get; set; }
27  virtual DbSet<Stagiaire> Stagiaire { get; set; }
28  virtual DbSet<Utilisateur> Utilisateur { get; set; }
29
30  override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
31  {
32      optionsBuilder.IsConfigured()
33
34      // Protect potentially sensitive information in your connection string, you should move it out of source code. See https://aka.ms/aspnetcore/sqlconn
35      optionsBuilder.UseSqlServer("Server=DESKTOP-23KGA17;Database=GestionStagiaire;Trusted_Connection=True;");
36
37  }
38
39  override void OnModelCreating(ModelBuilder modelBuilder)

```

Figure 12: Liaison avec la base de données

- Nous avons commencé après la création des différentes requêtes HTTP qui nous avons permis d'extraire les données à partir de la base donnée le formuler et l'envoyer vers le front-end à travers le serveur sous forme des fichiers JSON



```

44  [HttpGet("{id}")]
45  public async Task<ActionResult<EtatStagiaire>> GetEtatStagiaire(int id)
46  {
47      var etatStagiaire = await _context.EtatStagiaire.FindAsync(id);
48      _context.Entry(etatStagiaire)
49          .Collection(stage => stage.Stagiaire)
50          .Query()
51          .Include(stage => stage.Commentaire)
52          .Load();
53
54      if (etatStagiaire == null)
55      {
56          return NotFound();
57      }
58
59      return etatStagiaire;
60  }
61
62  // PUT: api/EtatStagiaires/5
63  // To protect from overposting attacks, enable the specific properties you want to bind to, for
64  // more details, see https://go.microsoft.com/fwlink/?linkid=2123754.
65  [HttpPut("{id}")]
66  public async Task<IActionResult> PutEtatStagiaire(int id, EtatStagiaire etatStagiaire)
67  {
68      if (id != etatStagiaire.IdEtatStagiaire)
69      {
70          return BadRequest();
71      }

```

Figure 13: Requêtes http

- Nous avons abordé de suite la phase de test à l'aide du Postman afin de corriger les erreurs qui ont eu lieu

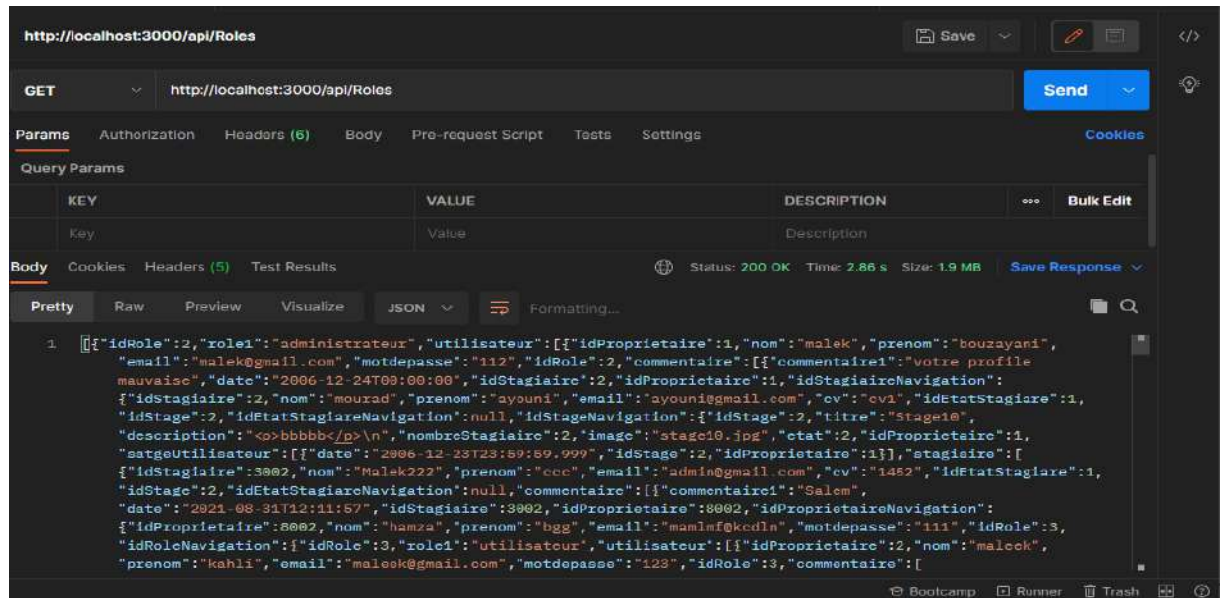


Figure 14: Test et correction avec Postman

5) Développement des Différents Interfaces du Front-end

- Nous avons Commencé par développer l'interface du login Ainsi que gérer les exceptions de l'authentification.

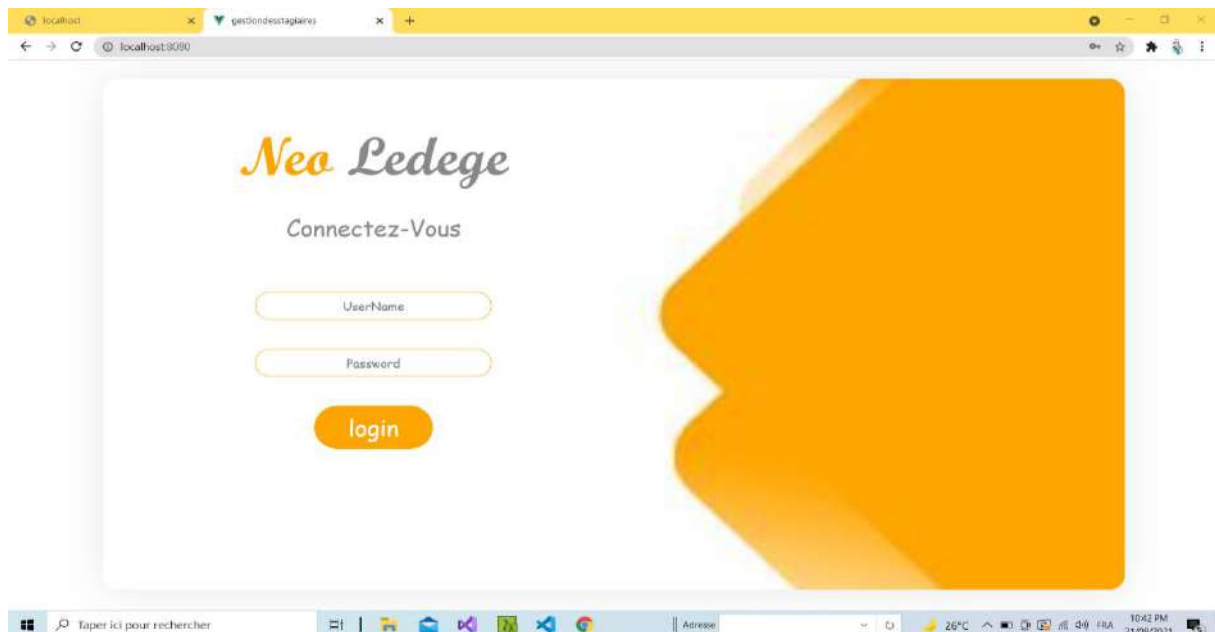


Figure 15: Interface du l'authentification

- Ensuite, nous avons passé au développement de l'interface d'accueil de l'utilisateur où il peut modifier, supprimer, éditer et ajouter les membres et voir les autres utilisateurs ayant accès à la plateforme.

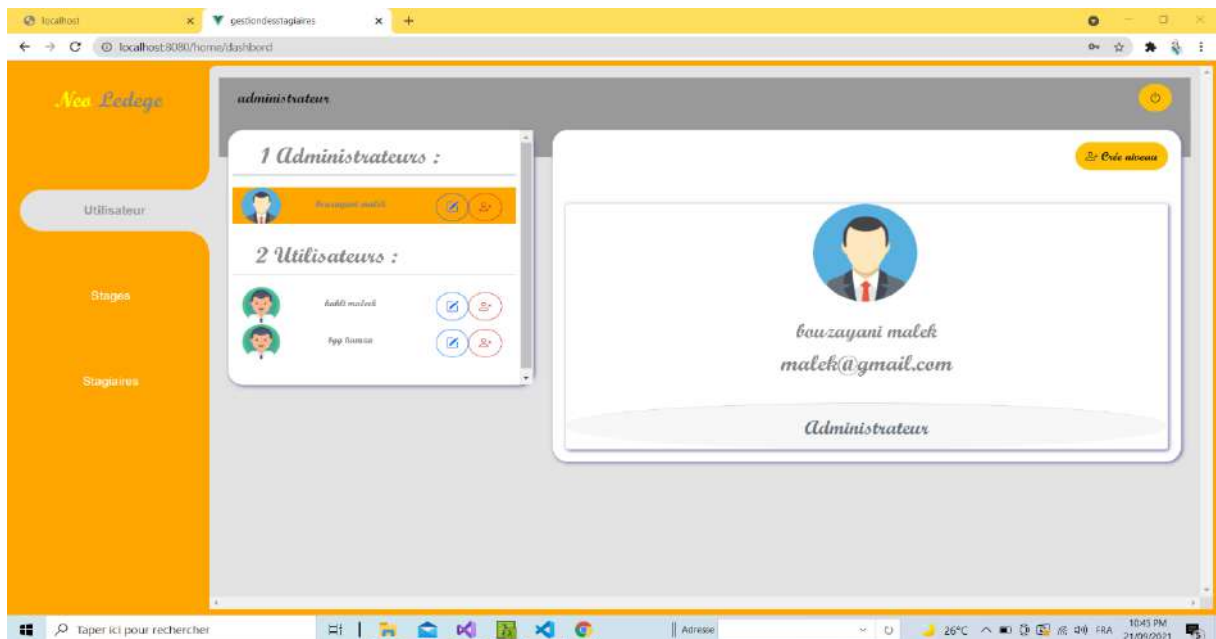


Figure 16: Interface d'accueil

- Nous avons ainsi conçu l'interface d'édit d'un membre où le responsable RH peut éditer un profil en spécifiant toutes les données demandées

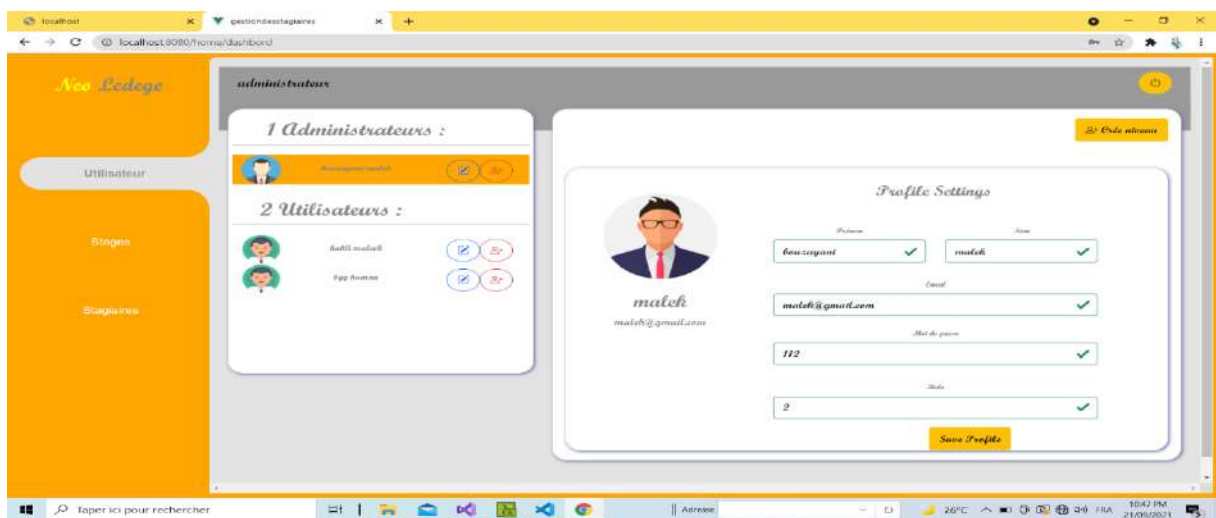


Figure 17: Interface édit utilisateur

- Nous avons développé l'interface d'ajout d'un membre où le responsable RH peut ajouter un nouveau membre en spécifiant toutes leurs informations.

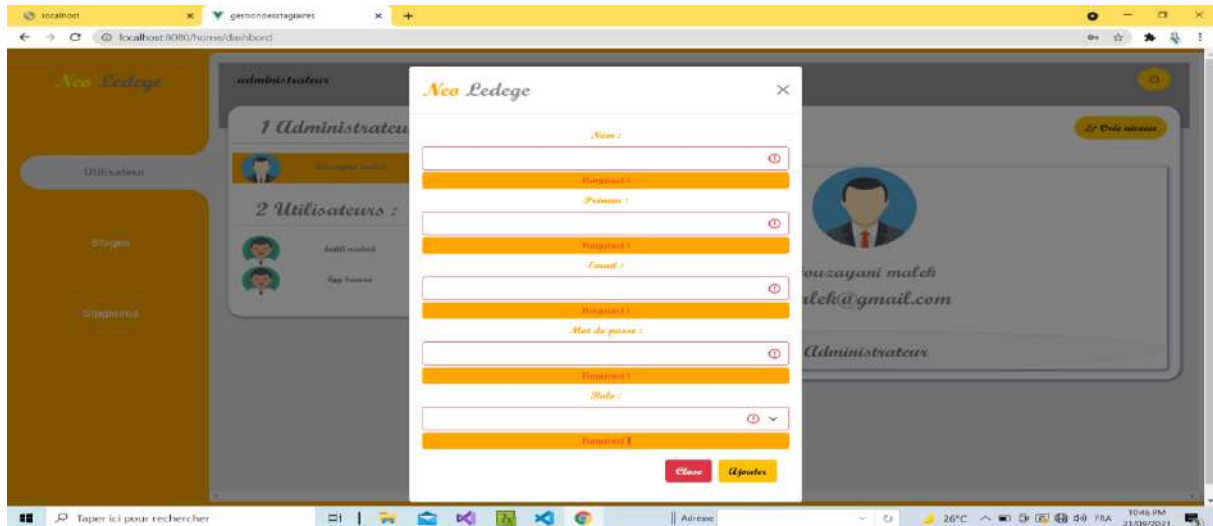


Figure 18: Interface ajoute utilisateur

- Puis nous avons ajouté la fonctionnalité de la suppression d'un membre qui se fait par le responsable Rh seulement.

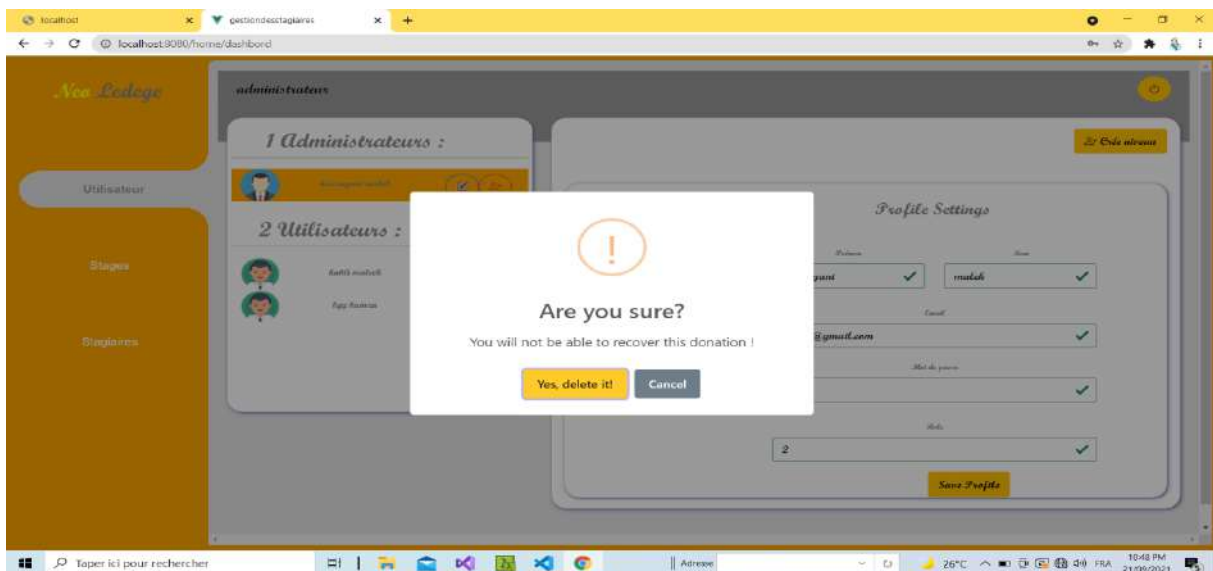


Figure 19: Interface supprimer utilisateur

- Par la suite, nous avons orienté vers le développement de l'interface qui illustre les différents stages et donnant un aperçu général sur chaque stage (Nombre de stagiaires, responsable ainsi que la disponibilité (affectée ou non))

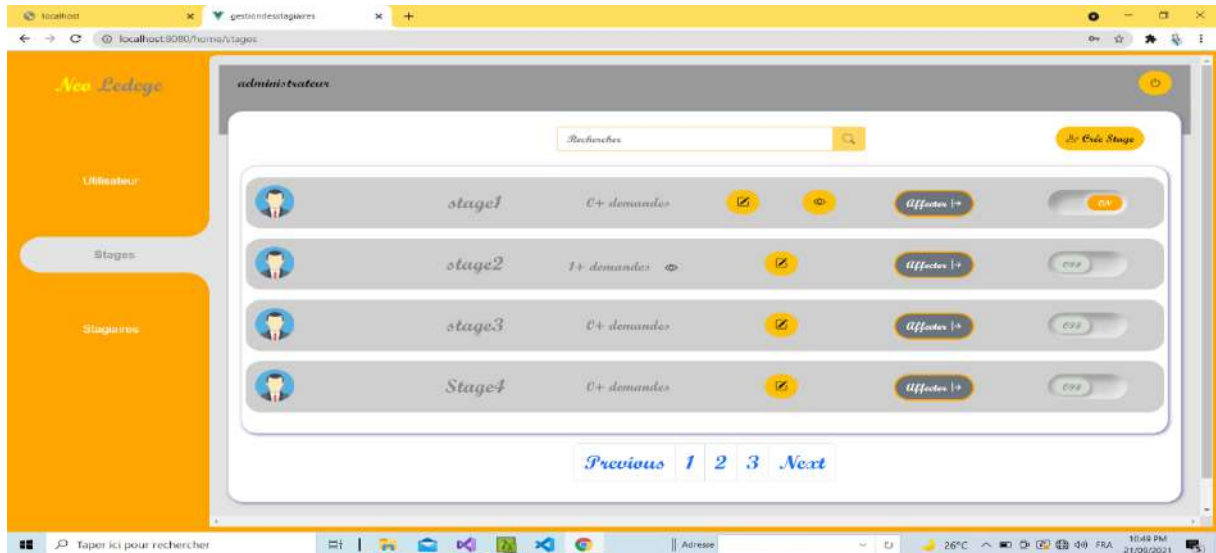


Figure 20: Interface liste des stages

- Nous avons développé l'interface d'ajout un stage où le responsable RH peut ajouter un nouveau stage en spécifiant toutes leurs informations.

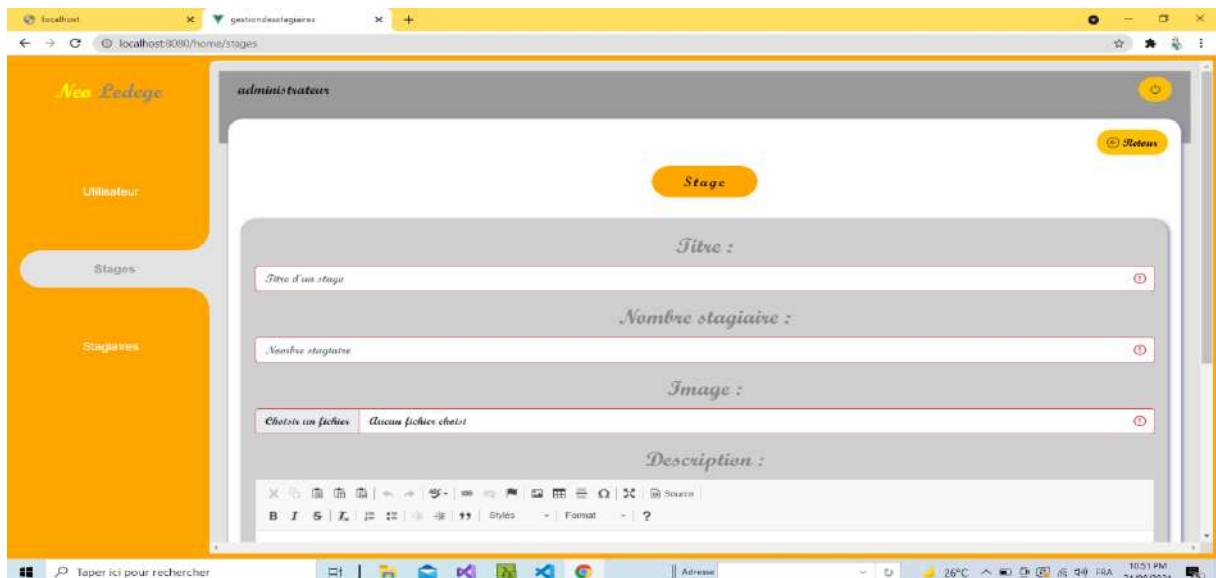


Figure 21 : Interface ajoute stage

- Puis nous avons ajouté la fonctionnalité de la recherche d'un stage par son titre.

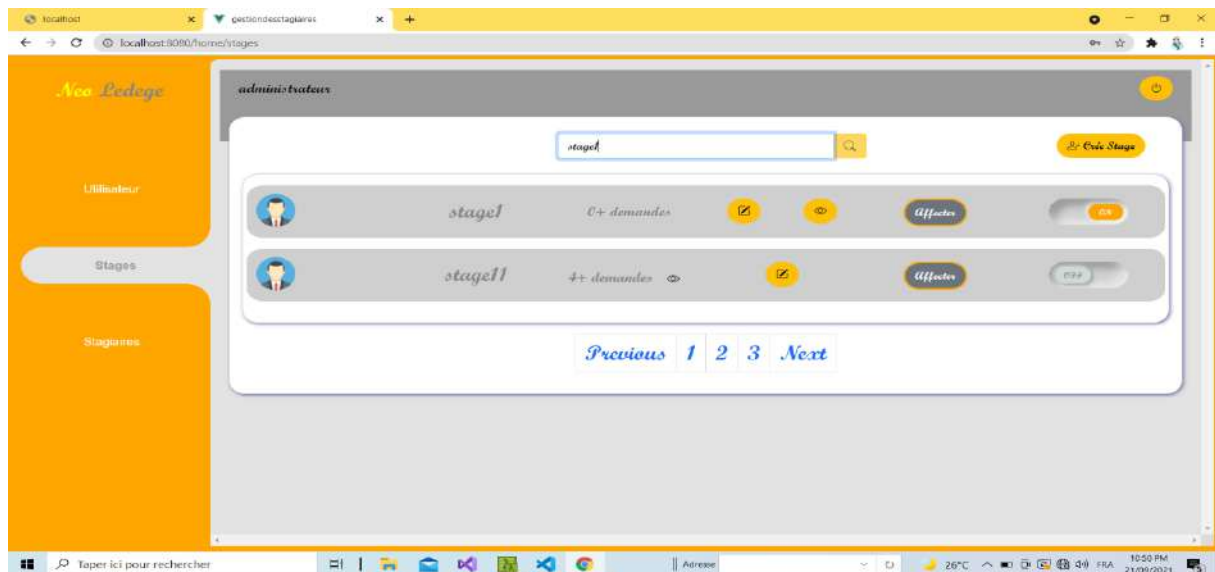


Figure 22: Chercher stage

- Nous avons ainsi conçu l'interface d'édit d'un stage.

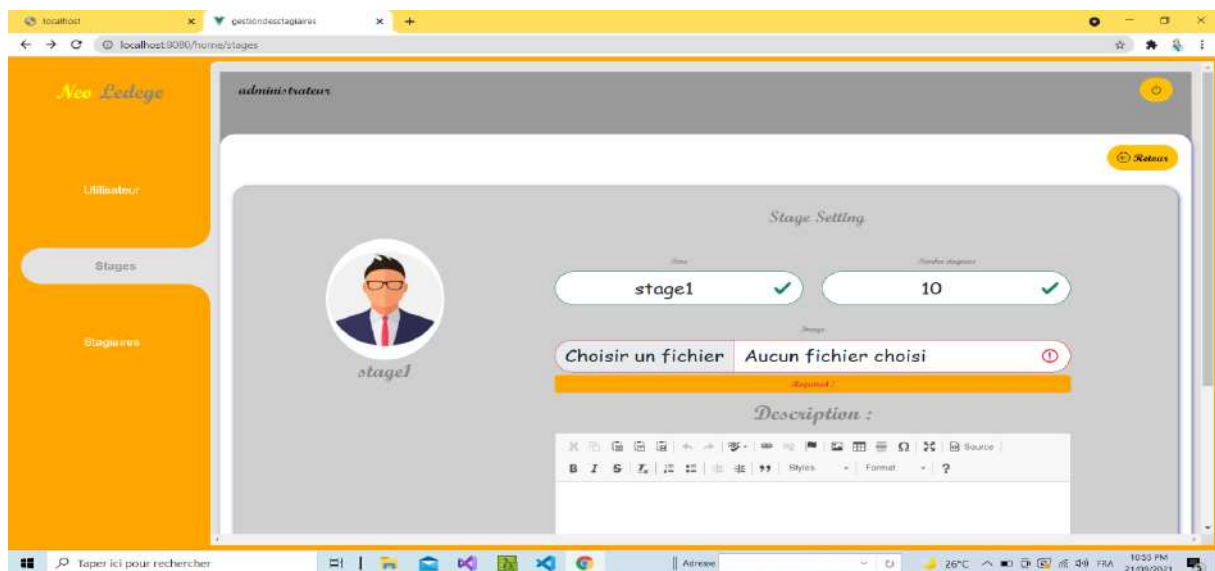


Figure 23: Interface édit stage

- Ensuite, nous avons développé l'interface de l'affectation d'un membre à un stage pour l'encadrer leur stagiaire.

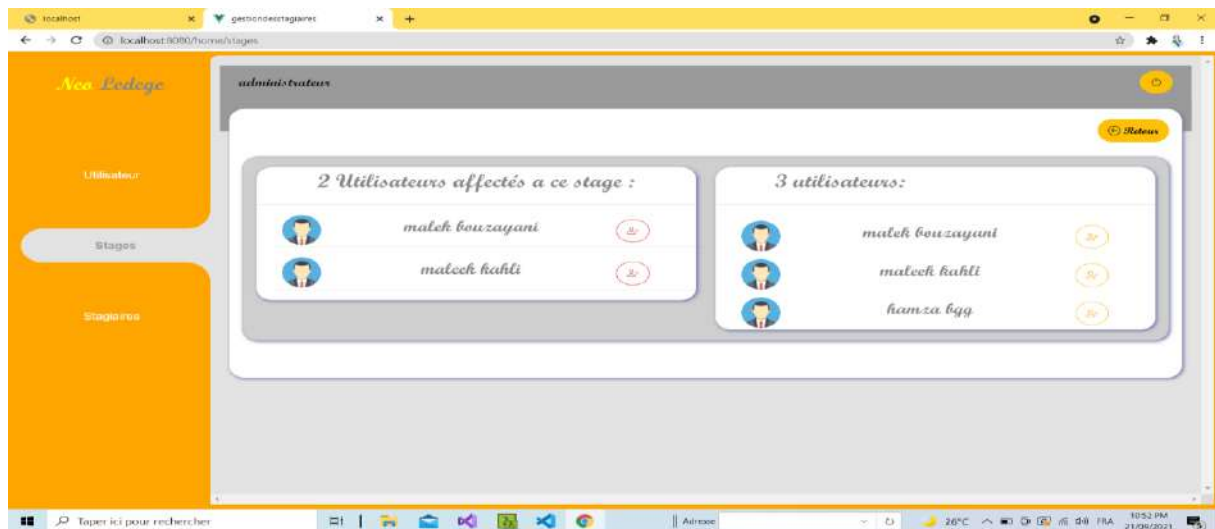


Figure 24 : Interface affecté utilisateur

- Ainsi, nous avons développé l'interface qui va publier et remplir par un candidat.



Figure 25: Interface publié stage

- Ensuite, nous avons conçu l'interface des stagiaires ou l'utilisateur aurait un aperçu sur les stagiaires qui ont postulé pour chacun, des stages et à travers lequel on peut affecter chaque stagiaire à un stage

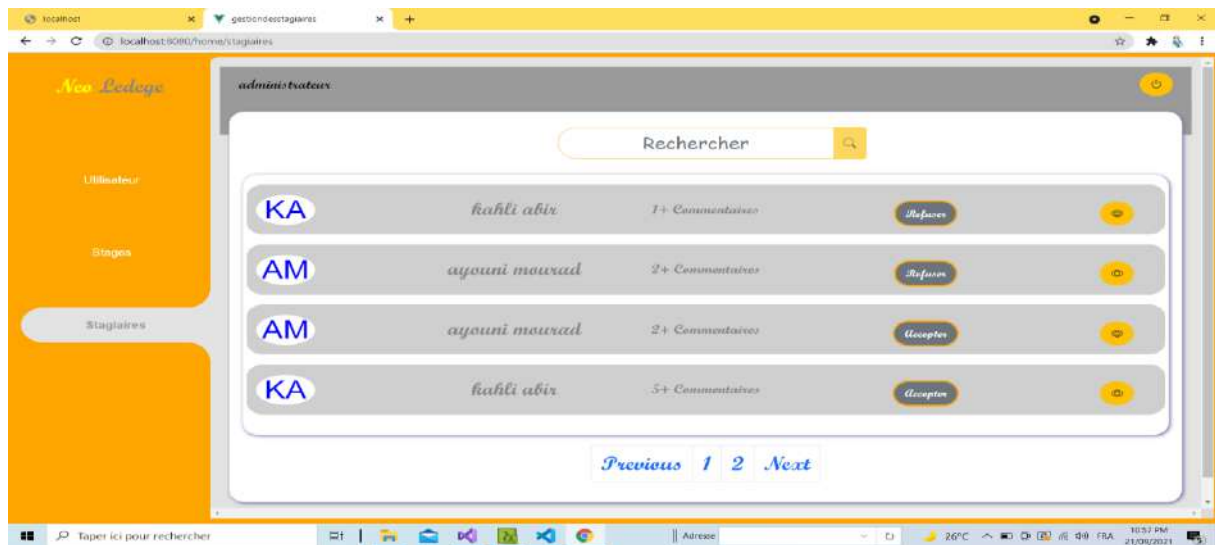


Figure 26: Interface liste stagiaire

- Puis, nous avons créé l'interface détail stagiaire pour que l'utilisateur voir les détails d'un stagiaire facilement et faire des commentaires.

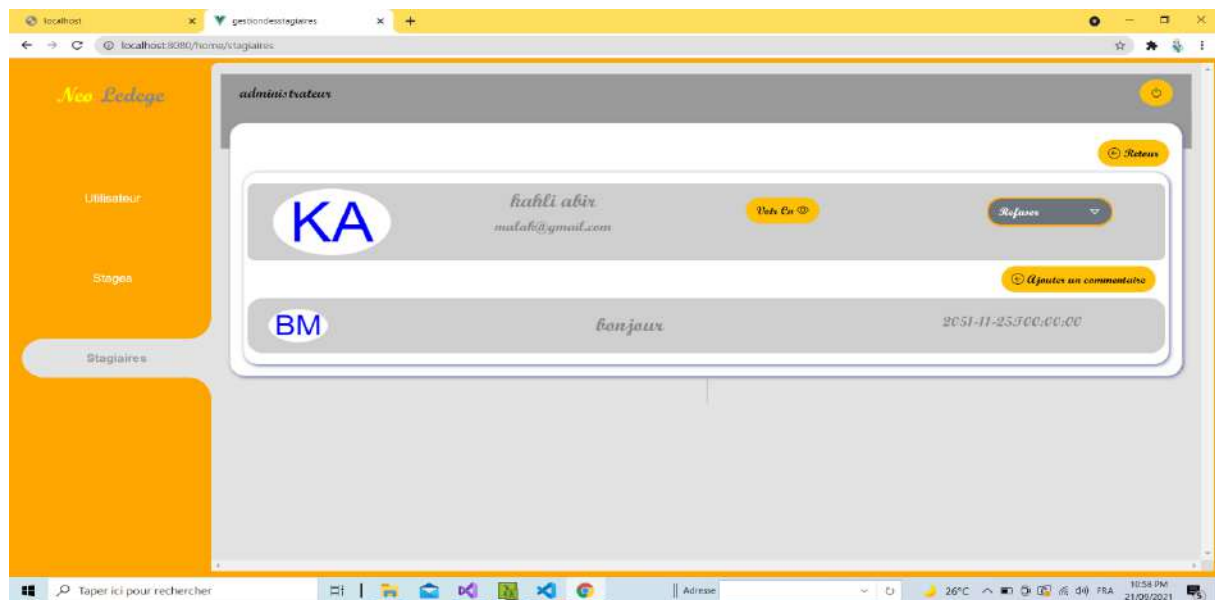


Figure 27: Interface détail stagiaire

- Ainsi nous avons ajouté la fonctionnalité ajoutée commentaire à l'interface détail stagiaire.

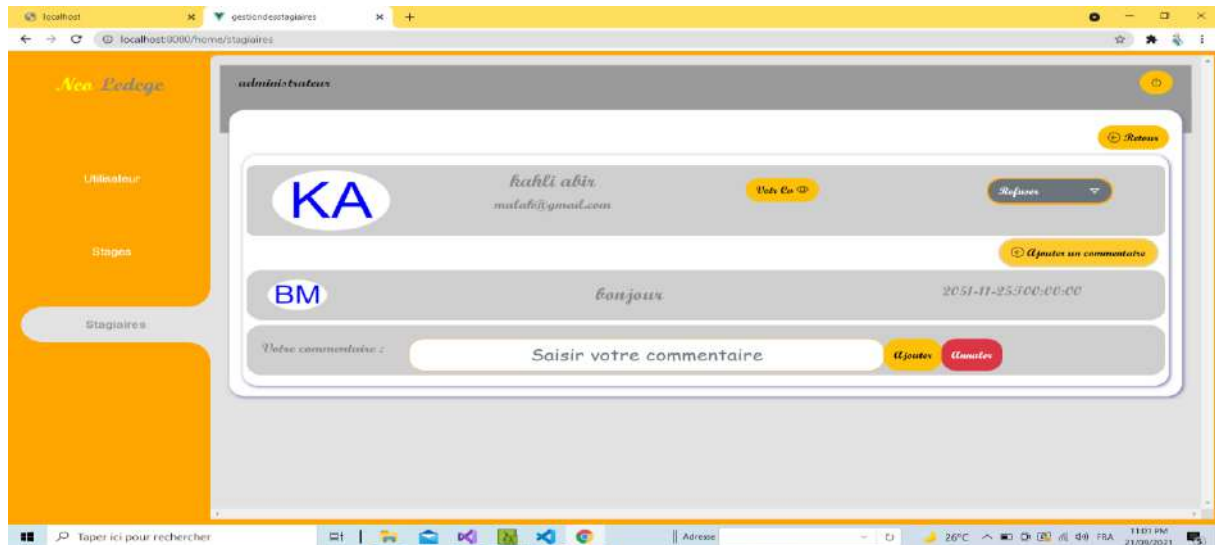


Figure 28: Interface ajouter commentaire

- Puis nous avons ajouté la fonctionnalité gérer l'état (refuser/accepter) d'un stagiaire.

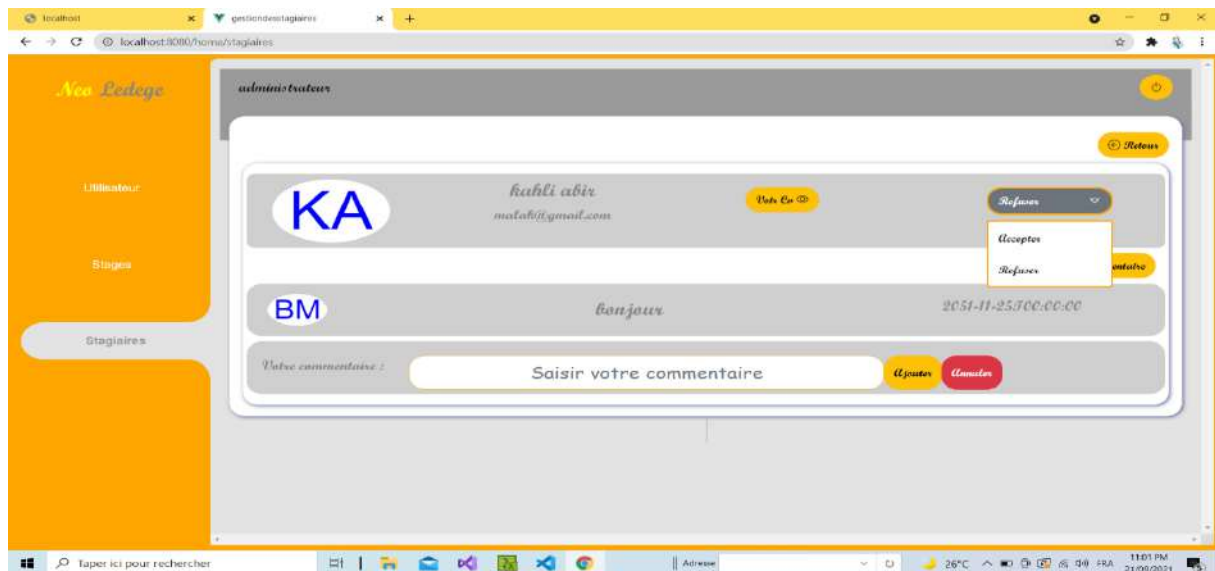


Figure 29: Interface accepté stagiaire

- Ainsi nous avons ajouté la fonctionnalité voir les stagiaires affecter à chaque stage.

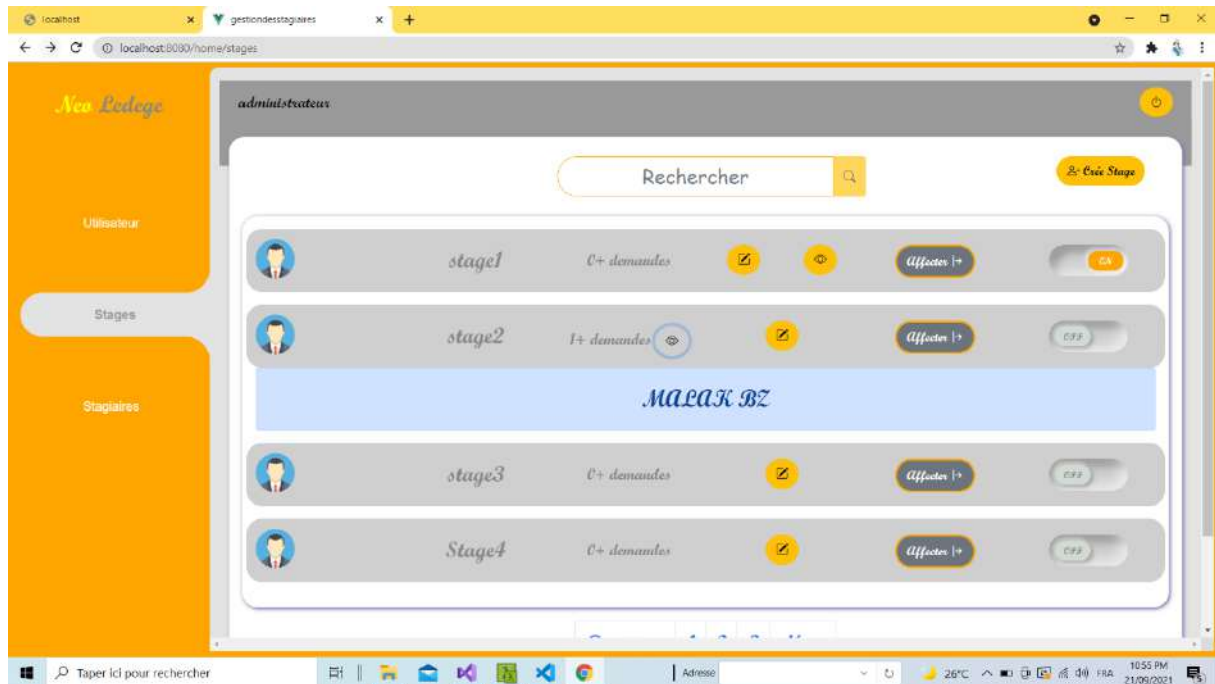


Figure 30 : Interface stagiaire accepter

6) Conclusion

Au bout de ce dernier chapitre nous avons abordé les différentes technologies utilisées lors de la phase de développement, ainsi nous avons présenté les différents résultats obtenus en introduisant certaines interfaces développées.

Conclusion

J'ai réalisé mon stage de perfectionnement au sein de Neolledge Station où ma mission était de développer une solution pour la gestion des stagiaires en utilisant les Framework VueJS et Asp. NetCore.

Ayant déjà des prérequis concernant le HTML et le JavaScript l'utilisation du VueJS n'était pas assez difficile quant au .NetCore développé en C#.

J'ai commencé le stage par une phase d'apprentissage afin de découvrir les notions de base de VueJS et .NetCore ensuite, j'ai passé vers la phase de développement au cours duquel j'ai maîtrisé de plus les notions acquies lors de la première phase, j'ai développé une Back-End à l'aide de .NetCore et Visual studio et ensuite j'ai commencé à développer les interfaces de l'utilisateur, ainsi j'ai utilisé Axios qui est un client http sur promesse pour consommer l'API au niveau du Front-End.

Bien que j'ai pu apprendre beaucoup de la vie normale d'un développeur, mon stage chez Neolledge Station a été un succès. J'ai pu acquérir des compétences pratiques, travailler dans un environnement fantastique et nouer des liens qui dureront toute ma vie. Je ne pourrais pas être plus satisfait.

Bibliographie

- Site officiel du VUEJS : <https://fr.vuejs.org/>
- JavaScript : <https://grafikart.fr/formations/debuter-javascript>
- L'essentiel du web dynamique avec ASP.NET :
 - ❖ <https://www.youtube.com/watch?v=1ck9LIBxO14>
 - ❖ <https://www.youtube.com/watch?v=BfEjDD8mWYg>
 - ❖ <https://www.youtube.com/watch?v=fmvcAzHpsk8>
 - ❖ <https://www.youtube.com/watch?v=-tZLsJEEqeU>
 - ❖ <https://www.youtube.com/watch?v=r4LIhYQ9GY>
- Documentation Asp. NetCore : <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>
- Axios : <https://fr.vuejs.org/v2/cookbook/using-axios-to-consume-apis.html>