

DOSSIER DE SOUTENANCE POUR LE TITRE DE CONCEPTEUR DÉVELOPPEUR APPLICATION

TABLE DES MATIÈRES

INTRODUCTION AU PROJET	4
1. Présentation personnelle	4
2. Présentation du projet en anglais	4
3. Compétences couvertes par le projet	5
ORGANISATION ET CAHIER DES CHARGES	6
1. Analyse de l'existant	6
2. Les utilisateurs du projet	6
3. Contexte technique	6
4. Les fonctionnalités attendues	6
1. Application mobile	6
2. Application web	6
CONCEPTION DU PROJET	7
1. Choix de développement	7
1. Choix des langages	7
2. Choix des frameworks	7
3. Logiciels et autres outils	7
2. Organisation du projet	7
3. Architecture logicielle	7
CONCEPTION DU FRONT-END DE L'APPLICATION	8
1. Arborescence du projet	8
2. Charte graphique	8
3. Maquettage	8
CONCEPTION DU BACK-END DE L'APPLICATION	9
1. La base de données	9
1. Concevoir une base de données	9
2. Mettre en place une base de données	9
3. Modèle conceptuel de données	9
4. Modèle logique de données	9
2. DÉVELOPPEMENT DU BACKEND DE L'APPLICATION	9
1. Organisation	9
2. Arborescence	9
3. Fonctionnement de l'API	9
4. Middleware	9
5. Routage	9
6. Controller	9
7. Service	9
8. Model	9

9. Sécurité	9
1. Chiffrement des données sensibles	9
2. JWT	9
3. Gestion des Droits	9
10. Problématique rencontrée	9
11. Problématique rencontrée	9
12. Recherches anglophones	9
13. Exemple :	9
14. Documentation	9
15. Tests	9
1. Postman	9
2. Newman	9
DÉVELOPPEMENT DU FRONT-END DE L'APPLICATION	11
1. Arborescence	11
2. Pages et composants	11
3. Sécurités	11
4. Problématiques rencontrées	11
5. Exemple navigation	11
6. Exemple de formulaire de mise à jour du profil	11
CONCEPTION DE L'ESPACE ADMINISTRATEUR	12
1. Conception de la partie administration	12
2. User Story	12
3. Choix du langage et framework	12
4. Conception du frontend du site web	12
1. Charte graphique	12
2. Maquettage	12
5. Conception du back end du site web	12
CONCLUSION	13
ANNEXES	14

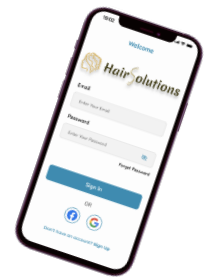
INTRODUCTION AU PROJET

1. Présentation personnelle

Je m'appelle Mériem BARKA. J'ai découvert la programmation suite à une formation de designer web que j'ai suivie avec ACOPAD; Cela m'a permis de m'ouvrir aux métiers liés au web et plus particulièrement le développement. J'aime créer et trouver des solutions aux problèmes rencontrés dans tous les domaines. J'ai suivi le cursus de la coding School à la plateforme en 2022 et j'ai obtenu mon titre de développeur web et web mobile. Aujourd'hui je suis en Coding School 2 afin de préparer mon titre de concepteur et développeur d'application web et étant en alternance au sein de l'entreprise DEKI.

2. Présentation du projet en anglais

HairSolution



We have developed a mobile application called "HairSolution" specifically designed to meet the needs of hair care enthusiasts.

We have observed that many people are in need of advice and guidance for their hair care.

Whether you are seeking personalized advice, solutions for specific hair issues, or simply engaging discussions with other hair care enthusiasts, our application is the ideal tool to guide you in your hair care routine.

With Hair Solution, you can expand your knowledge of hair care by joining a dynamic and committed community.

Share your experiences, ask questions, receive expert advice, and discover new tips to improve the health of your hair.

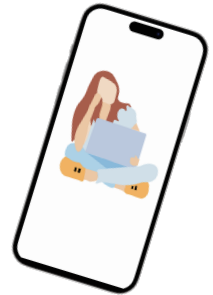
Our application provides a friendly space where users can interact with each other, exchange tips, and discuss the latest trends in hair care.

Download Hair Solution now and join a passionate and caring community!

3. Compétences couvertes par le projet

Ce projet couvre les compétences du titre suivantes :

- Maquetter une application
- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur web
- Développer la partie back-end d'une interface utilisateur web
- Concevoir une base de données
- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données
- Concevoir une application
- Développer des composants métier
- Construire une application organisée en couches
- Développer une application mobile
- Préparer et exécuter les plans de tests d'une application
- Préparer et exécuter le déploiement d'une application



ORGANISATION

Nous étions deux dans le groupe et nous nous sommes réparti les tâches de la conception à la livraison du projet. Nous nous sommes organisés de la manière suivante :

1. Définition du projet :

- Nous avons discuté ensemble de l'objectif et des spécifications du site web que nous souhaitons créer.
- Nous avons identifié les fonctionnalités clés et les exigences techniques.

2. Conception :

- On préfère le faire ensemble pour la conception graphique et l'élaboration de la charte graphique.
- Ainsi que sur la conception de l'architecture de l'information et la création des wireframes.

3. Développement :

- J'ai assumé la responsabilité du développement front-end, en utilisant react Native, expo.. pour créer l'interface utilisateur.
- J'ai pris en charge le développement back-end, "les routes" en utilisant des langages de programmation tels que node js, express pour mettre en place la logique et les fonctionnalités du site.

4. Intégration et tests :

- Nous avons collaboré étroitement pour intégrer les éléments front-end et back-end et nous assurer que tout fonctionnait correctement.
- Nous avons effectué des tests réguliers pour détecter et corriger les bugs et les problèmes de performance.

5. Révision et finalisation :

- Nous avons revu ensemble le site web dans son ensemble, en vérifiant la cohérence visuelle, la fluidité de la navigation et le bon fonctionnement de toutes les fonctionnalités.
- Nous avons apporté les dernières modifications nécessaires avant de considérer le projet comme terminé.

6. Livraison :

- Ensemble, nous avons préparé les fichiers et les ressources nécessaires pour la mise en ligne du site web.
- Nous avons coordonné la mise en place du site sur un serveur ou une plateforme d'hébergement.

Dans cette organisation à deux personnes, nous avons tiré parti de nos compétences complémentaires pour mener à bien toutes les phases du projet, en assurant une communication régulière et une répartition équilibrée des responsabilités.

CAHIER DES CHARGES

1. Analyse de l'existant

Nous avons effectué des recherches, mais nous n'avons pas trouvé d'application mobile qui propose ce que nous souhaitons offrir aux utilisateurs.

2. Les utilisateurs du projet

Bienvenue dans l'application mobile dédiée aux soins capillaires ! Notre objectif principal est d'offrir aux utilisateurs une expérience enrichissante en lui fournissant des astuces capillaires de qualité. Cette application est destinée aux hommes et aux femmes, jeunes ou plus âgé(e)s, quels que soient vos types de cheveux (bouclés, lisses, frisés, longs ou courts), que vous cherchiez à résoudre des problèmes spécifiques liés à vos cheveux ou simplement à découvrir de nouvelles astuces pour les sublimer.

Nous sommes là pour vous accompagner dans votre parcours capillaire.

Nous souhaitons vous permettre d'améliorer vos connaissances sur les soins capillaires en vous donnant accès à des astuces pratiques adaptées à chaque type de cheveux, des conseils pour prévenir les pointes fourchues et des routines capillaires personnalisées.

L'application mobile vise à créer une communauté dédiée à tous ceux et celles qui souhaitent échanger des astuces, conseils et routines pour prendre soin de leurs cheveux.

Nous encourageons l'échange et l'entraide entre les utilisateurs, vous pourrez ainsi partager vos propres astuces et découvertes.

3. Contexte technique

Pour une application mobile de soins capillaires, le contexte technique inclut les éléments suivants :

1. **Plateforme** : on a décidé de développer l'application mobile sur plateforme iOS, et Android (multiplateforme).
2. **Langage de programmation** : Le choix du langage de programmation que nous avons choisi est react native, car il offre plusieurs avantages significatifs :
 - **Multiplateforme** : React Native permet de développer une seule base de code qui fonctionne à la fois sur iOS et Android. Cela réduit considérablement les efforts de développement et de maintenance, car une grande partie du code peut être partagée entre les deux plateformes.

- **JavaScript** : React Native utilise JavaScript, l'un des langages de programmation les plus populaires et répandus. Il est largement connu et maîtrisé par de nombreux développeurs, ce qui facilite le recrutement et la collaboration au sein de l'équipe de développement.
- **Écosystème React** : React Native est basé sur React, une bibliothèque JavaScript bien établie pour la création d'interfaces utilisateur. Cela signifie que les développeurs qui connaissent déjà React pour le développement Web peuvent facilement transférer leurs compétences pour créer des applications mobiles avec React Native.
- **Performances** : Bien que React Native ne soit pas aussi rapide que le développement natif (Kotlin pour Android, Swift pour iOS), il se rapproche beaucoup des performances natives. React Native utilise des composants natifs pour les parties critiques de l'application, ce qui améliore considérablement les performances par rapport aux frameworks hybrides traditionnels.
- **Communauté active** : React Native bénéficie d'une large communauté de développeurs qui contribuent à son amélioration continue. Cela signifie qu'il existe de nombreuses bibliothèques open-source, des modules complémentaires et des ressources disponibles pour accélérer le développement.
- **Rapidité de développement** : Le fait de partager une grande partie du code entre les plateformes permet d'accélérer le processus de développement. Les équipes peuvent donc créer des applications plus rapidement, ce qui est un avantage essentiel dans le domaine des technologies mobiles.
- **Hot Reloading** : React Native propose une fonctionnalité appelée "Hot Reloading" qui permet aux développeurs de voir instantanément les changements apportés au code sans avoir à recompiler l'application. Cela facilite le processus de débogage et améliore la productivité des développeurs.

3. Environnement de développement nous avons intégré L'IDE Visual Studio Code c'est un IDE dédié pour écrire, tester et déboguer le code de l'application.
Les IDE populaires incluent Xcode pour iOS et Android Studio pour Android.
4. Conception de l'interface utilisateur (UI) / Expérience utilisateur (UX) : La conception de l'interface utilisateur a été réalisée en accord avec les principes de l'expérience utilisateur pour créer une application attrayante et facile à utiliser.
5. Base de données : Pour stocker les informations telles que les profils des utilisateurs, les astuces capillaires, les préférences des utilisateurs, etc., l'application mobile nécessite une base de données.
Nous avons opté pour l'utilisation d'une base de données NoSQL car cette technologie offre une grande flexibilité et évolutivité.
Avec une base de données NoSQL, nous pouvons gérer facilement des données non structurées ou semi-structurées, ce qui correspond parfaitement aux besoins variables d'une application de soins capillaires.

De plus, la capacité de mise à l'échelle horizontale des bases de données NoSQL nous permettra de gérer efficacement la croissance future de l'application et d'assurer des performances optimales, même avec un grand nombre d'utilisateurs et de données.

6. API : Si l'application se connecte à un serveur ou à des services tiers, des API (Interfaces de Programmation Applicative) seront utilisées pour faciliter les échanges de données.
7. Sécurité : Des mesures de sécurité ont été mises en place pour protéger les données des utilisateurs, telles que le chiffrement des données, l'authentification sécurisée, etc.
8. Notifications push : Pour envoyer des notifications aux utilisateurs, il faudra mettre en place un système de notifications push.
9. Test et débogage : Des tests rigoureux ont été effectués pour identifier et corriger les bugs avant le lancement de l'application.
avec l'outil postman et le console log pour déboguer ou pour résoudre les problèmes techniques éventuels.
10. Déploiement : Une fois l'application prête, elle sera soumise aux magasins d'applications (App Store pour iOS, Google Play Store pour Android) pour être disponible en téléchargement.
11. Maintenance : Après le lancement, l'application nécessitera des mises à jour régulières pour ajouter de nouvelles fonctionnalités, corriger les bugs et s'adapter aux évolutions des plateformes mobiles.

12. Les fonctionnalités attendues

1. Application mobile

Voici les fonctionnalités attendues dans une application mobile de tchat et forum dédiée aux soins et astuces capillaires :

Tchat en direct : Une fonctionnalité de messagerie instantanée qui permet aux utilisateurs d'échanger des messages en direct avec d'autres membres de la communauté.

Forum de discussion : Un espace où les utilisateurs peuvent créer des sujets de discussion, poser des questions, partager des astuces et des expériences, et interagir avec d'autres passionnés de soins capillaires.

Partage d'astuces et conseils : Les utilisateurs peuvent publier leurs propres astuces, conseils et routines capillaires pour les partager avec la communauté.

Gestion de profil utilisateur : Les utilisateurs peuvent créer et personnaliser leur profil, ajouter des informations sur leur type de cheveux, leurs préférences capillaires, etc.

Notifications personnalisées : Les utilisateurs peuvent choisir de recevoir des notifications sur les nouveaux messages, les réponses à leurs publications ou les sujets qui les intéressent.

Suivi des sujets favoris : Les utilisateurs peuvent marquer des sujets ou des discussions comme favoris pour y accéder facilement ultérieurement.

Système de likes et de commentaires : Les utilisateurs peuvent liker les publications d'autres membres et laisser des commentaires pour engager des discussions.

Recherche avancée : Une fonctionnalité de recherche qui permet aux utilisateurs de trouver des sujets spécifiques, des astuces ou des membres de la communauté.

Modération et rapport : Un système de modération pour gérer les contenus inappropriés et la possibilité pour les utilisateurs de signaler des publications problématiques.

Synchronisation des données : Les données de l'application, telles que les discussions et les profils des utilisateurs, peuvent être synchronisées pour une utilisation hors ligne.

En intégrant ces fonctionnalités de tchat et de forum, l'application mobile permettra aux utilisateurs d'échanger des informations, de partager leurs connaissances et de bénéficier d'une communauté engagée dans les soins et astuces capillaires.

2. Application web

Voici les fonctionnalités attendues pour l'application web destinée à l'administration et à la gestion de l'application dédiée aux soins capillaires :

1. **Tableau de bord d'administration :** Un tableau de bord convivial et sécurisé pour les administrateurs, offrant un aperçu des principales activités et statistiques de l'application.
2. **Gestion des utilisateurs :** La possibilité pour les administrateurs de gérer les comptes des utilisateurs, d'ajouter de nouveaux utilisateurs, de modifier des profils et de gérer les autorisations.
3. **Modération des contenus :** Un système de modération qui permet aux administrateurs de surveiller et de modérer les publications, les commentaires et les discussions pour assurer le respect des règles de la communauté.

-
4. Gestion des astuces et des tutoriels : Les administrateurs peuvent ajouter, éditer ou supprimer des astuces, des conseils et des tutoriels pour les partager avec les utilisateurs.
 5. Gestion du contenu du forum : La possibilité de gérer les sujets de discussion, d'approuver ou de supprimer des publications et de surveiller l'activité du forum.
 6. Statistiques et analyses : Un système d'analyse qui fournit des rapports sur l'utilisation de l'application, l'engagement des utilisateurs, les sujets les plus populaires, etc.
 7. Gestion des notifications : La possibilité de gérer les notifications envoyées aux utilisateurs, y compris les notifications push et les e-mails.
 8. Système d'authentification et de sécurité : Un système robuste d'authentification et de sécurité pour protéger l'accès aux fonctionnalités d'administration.
 9. Gestion des produits et partenariats : Si l'application propose des produits capillaires recommandés ou des partenariats avec des marques, les administrateurs peuvent gérer ces éléments.
 10. Gestion des rapports d'utilisateurs : La possibilité de gérer les rapports soumis par les utilisateurs concernant des contenus inappropriés ou des problèmes techniques.
 11. Système de sauvegarde et de restauration : Un mécanisme de sauvegarde régulière des données de l'application pour la prévention des pertes de données et la possibilité de restaurer des sauvegardes si nécessaire.
 12. Paramètres du site : Les administrateurs peuvent gérer les paramètres globaux de l'application, tels que les paramètres d'affichage, les options de langue, etc.

En intégrant ces fonctionnalités pour l'administration de l'application web, les administrateurs pourront gérer efficacement l'ensemble de l'application dédiée aux soins capillaires, assurer la qualité du contenu et offrir une expérience utilisateur optimale aux utilisateurs de la communauté.

CONCEPTION DU PROJET

1. Choix de développement

On a choisi Le d'utiliser MERN pour le développement est une excellente décision, car cette technologie est populaire et puissante pour la création d'applications web modernes.

MERN est un acronyme qui représente les technologies suivantes :

1. MongoDB : Une base de données NoSQL orientée documents, très adaptée pour stocker des données JSON.
2. Express : Un framework JavaScript pour le développement d'applications web côté serveur basées sur Node.js. Express facilite la création de routes, la gestion des requêtes et des réponses, ainsi que la configuration du serveur.
3. React : Une bibliothèque JavaScript pour la création d'interfaces utilisateur interactives et réactives. React permet de construire des composants réutilisables qui facilitent la création d'expériences utilisateur dynamiques.
4. Node.js : Un environnement d'exécution JavaScript côté serveur qui permet d'utiliser JavaScript pour créer des applications web côté serveur.

En combinant ces quatre technologies, MERN offre une stack complète pour le développement web full-stack en JavaScript. Cela permet une meilleure cohérence du code entre le côté client et le côté serveur, ce qui peut accélérer le processus de développement.

1. Choix des langages

Nous avons fait le choix d'utiliser les langages suivants : ReactJS, React Native et Node.js est cohérent et complémentaire pour le développement d'applications web et mobiles complètes et réactives.

1. ReactJS : React est une bibliothèque JavaScript pour la création d'interfaces utilisateur interactives et réactives côté client. Avec React, vous pouvez construire des composants réutilisables qui facilitent la création d'une expérience utilisateur fluide et dynamique pour les applications web.
2. React Native : React Native est une extension de React qui permet de développer des applications mobiles natives pour iOS et Android en utilisant JavaScript. Grâce à React Native, vous pouvez partager une grande partie du code entre les plateformes tout en offrant une expérience utilisateur proche de celle des applications natives.

3. Node.js : Node.js est un environnement d'exécution côté serveur basé sur JavaScript. Il permet d'utiliser JavaScript pour développer des applications web côté serveur. Node.js est connu pour sa rapidité et son évolutivité, ce qui le rend idéal pour gérer des applications en temps réel et des API.

Ces langages sont tous basés sur JavaScript, ce qui permet une expérience de développement unifiée, car l'équipe n'a besoin de maîtriser qu'un seul langage pour l'ensemble du projet.

2. Choix des frameworks

Nous avons fait le choix d'utiliser le framework Express est une décision solide pour le développement d'applications web côté serveur basées sur Node.js. Express est l'un des frameworks les plus populaires et les plus largement utilisés dans l'écosystème Node.js en raison de sa simplicité, de sa flexibilité et de sa robustesse.

Voici quelques raisons pour lesquelles on a choisie Express :

1. Simplicité : Express est conçu pour être simple et facile à utiliser. Sa syntaxe claire et sa mise en œuvre intuitive permettent aux développeurs de démarrer rapidement et de créer des applications sans trop de complexité.
2. Flexibilité : Express est très flexible, ce qui signifie que vous pouvez choisir comment structurer votre application et quelles fonctionnalités ou middleware intégrer selon vos besoins spécifiques.
3. Large adoption et communauté active : Express est l'un des frameworks les plus populaires, avec une large adoption parmi la communauté des développeurs. Cela signifie qu'il existe de nombreuses ressources, des tutoriels, des plugins et des middleware tiers disponibles pour faciliter le développement.
4. Middleware : L'une des forces d'Express réside dans son système de middleware. Les middleware sont des fonctions qui peuvent être utilisées pour effectuer des tâches telles que l'authentification, la gestion des erreurs, la journalisation, etc. Ils offrent une flexibilité supplémentaire pour personnaliser le comportement de votre application.
5. Performance : Express est bien connu pour ses performances élevées, ce qui en fait un choix approprié pour les applications en temps réel et les API.

6. **Large écosystème** : Express est soutenu par un écosystème riche qui comprend de nombreux modules, bibliothèques et outils complémentaires pour étendre ses fonctionnalités.

Express : Express est un framework JavaScript pour Node.js qui simplifie le développement d'applications web côté serveur. Il fournit des fonctionnalités de routage, de gestion des requêtes et des réponses, ainsi que la possibilité d'utiliser des middleware pour gérer des tâches spécifiques comme l'authentification et l'autorisation.

3. Logiciels et autres outils

Nous avons fait le choix d'utiliser les outils suivants qui sont pertinents pour le développement et la gestion de projets numériques. Chacun de ces outils offre des fonctionnalités uniques et complémentaires pour faciliter le travail d'équipe, la gestion de projets et la conception d'interfaces utilisateur.

avantages de chaque outil :

1. **GitHub** : GitHub est un outil essentiel pour la collaboration dans le développement de logiciels. Il permet de gérer le code source de manière centralisée, de suivre les changements, d'effectuer des révisions de code (pull requests), de gérer les problèmes et les demandes de fonctionnalités, et de faciliter la collaboration entre les développeurs. GitHub offre une plateforme robuste pour la gestion du code, ce qui facilite le travail en équipe et la contribution à des projets open source.
2. **Trello** : Trello est un outil de gestion de projet basé sur des tableaux Kanban qui permet d'organiser et de suivre les tâches de manière visuelle. Il offre une vue d'ensemble de l'état d'avancement du projet, permettant à l'équipe de suivre les tâches en cours, les tâches terminées et celles en attente. Trello est idéal pour la planification des projets, le suivi des tâches et la collaboration entre les membres de l'équipe.
3. **Figma** : Figma est un outil de conception et de prototypage d'interfaces utilisateur collaboratif basé sur le cloud. Il permet aux concepteurs de travailler ensemble en temps réel, de créer des maquettes de conception interactives et de partager facilement des conceptions avec les parties prenantes. Figma offre une approche fluide de la conception et permet de collaborer efficacement pour créer des interfaces utilisateur attrayantes et conviviales.
4. **Canva** : Canva est un outil de conception graphique en ligne convivial, idéal pour créer des graphiques, des présentations, des publications sur les réseaux sociaux, des affiches et bien plus encore. Avec ses modèles prédéfinis et ses bibliothèques d'images, Canva facilite la création de contenus visuels de qualité, même pour les personnes n'ayant pas de compétences en conception.

En combinant ces outils, nous avons pu bien travailler en collaboration, gérer les projets de manière organisée et créer des interfaces utilisateur et des contenus visuels attrayants. Cette combinaison complète vous permettra de gagner du temps, d'améliorer la communication et la productivité, et de fournir des résultats de haute qualité dans le développement et la gestion de vos projets numériques.

2. Organisation du projet

Pour l'organisation du projet nous nous sommes organisés de manière suivante :

Chacune de nous a d'abord essayé de prendre en main les nouvelles technologies pour pouvoir comprendre et apprendre cette nouvelle technologie.

Lorsque nous avons adopté cette nouvelle **stack** technologique.

Pour nous organiser nous avons utilisé des outils tels que **GitHub** et **Trello** pour faciliter notre travail **collaboratif** et améliorer notre productivité.

Nous avons également utilisé **Figma** pour la **conception** et le **prototypage**, ce qui nous a permis de collaborer de manière fluide sur l'aspect visuel de notre projet.

GitHub a été notre principal outil pour la gestion du code source. Nous avons créé un dépôt commun pour le projet, où nous avons une branche "main" représentant la version stable du code.

Chacun de nous avait également sa propre branche de développement, nommée "dev", où nous pouvions travailler individuellement sur des fonctionnalités spécifiques ou des corrections sans perturber le code principal.

Pour organiser notre travail et suivre notre progression, nous avons utilisé Trello. Nous avons créé un tableau Trello dédié au projet, avec des listes représentant différentes étapes du développement telles que "à faire", "en cours" et "terminé".

Chaque tâche à accomplir était représentée par une carte dans le tableau, et nous pouvions les déplacer d'une liste à une autre au fur et à mesure de notre avancement.

Cela nous a permis de garder une vue d'ensemble de toutes les tâches et de nous assurer que rien n'était oublié.

Concernant notre environnement de développement, nous avons utilisé une nouvelle stack technologique pour ce projet.

Nous avons choisi **MongoDB** pour la gestion de la base de données, **React Native** et **Expo** pour le développement mobile, et **React.js** avec **Express** pour le développement Web. Pour interagir avec MongoDB, nous avons utilisé Mongoose pour faciliter la modélisation des données et les opérations **CRUD**.

Malgré que nous étions que deux dans l'équipe, nous avons réussi grâce à une collaboration étroite et une communication régulière.

En utilisant les outils GitHub et Trello, ainsi que la nouvelle stack technologique, nous avons développé efficacement notre projet tout en apprenant et progressant ensemble.

Notre approche collaborative nous a permis de surmonter les défis et de répondre aux attentes de l'équipe et des utilisateurs.

Nous avons pu travailler de manière autonome tout en partageant nos connaissances sur GitHub et en coordonnant nos efforts via Trello. Cette approche nous a permis de progresser rapidement dans l'apprentissage de la nouvelle technologie tout en maintenant une communication et une coordination efficaces au sein de l'équipe.

3. Architecture logicielle

L'architecture logicielle est la structure fondamentale et la conception globale d'un système logiciel ou d'une application informatique de ce fait nous nous somme bien centré sur cette partie du projet. Pour définir de manière propre on a choisi de faire du MVC (Model,View,Controller) en séparant les différents composants du logiciel.

Les composants sont organisés de manière à qu'ils interagissent entre eux et communiquent pour accomplir les tâches spécifiques du programme.

L'objectif principal de l'architecture logicielle est de créer une base solide pour le développement, la maintenance et l'évolutivité du logiciel.

La Scalabilité : D'une bonne architecture logicielle permet à l'application de s'adapter facilement aux changements de charge et de croissance en termes de volume d'utilisateurs ou de données.

Dans l'architecture logicielle, on cherche à bien séparer les différentes parties du logiciel (par exemple, la partie qui gère les règles métier de celle qui s'occupe de l'interface utilisateur). Cela permet de rendre le code plus clair et plus facile à comprendre, ce qui facilite aussi sa maintenance.

Modularité : L'architecture logicielle favorise la conception modulaire du logiciel, ce qui permet de réutiliser des composants et de faciliter le travail collaboratif entre les développeurs.

Une architecture logicielle bien conçue est essentielle pour le succès et la durabilité d'un projet logiciel. Elle aide à éviter les problèmes de maintenance, à améliorer la qualité du code et à faciliter l'ajout de nouvelles fonctionnalités tout au long du cycle de vie du logiciel.

CONCEPTION DU FRONT-END DE L'APPLICATION

1. Arborescence du projet
2. Charte graphique
3. Maquettage

CONCEPTION DU BACK-END DE L'APPLICATION

1. La base de données

1. Concevoir une base de données
2. Mettre en place une base de données
3. Modèle conceptuel de données
4. Modèle logique de données

2. DÉVELOPPEMENT DU BACKEND DE L'APPLICATION

1. Organisation
2. Arborescence
3. Fonctionnement de l'API
4. Middleware
5. Routage
6. Controller
7. Service
8. Model
9. Sécurité
 1. Chiffrement des données sensibles
 2. JWT
 3. Gestion des Droits
10. Problématique rencontrée
11. Problématique rencontrée
12. Recherches anglophones
13. Exemple :
14. Documentation
15. Tests
 1. Postman
 2. Newman

DÉVELOPPEMENT DU FRONT-END DE L'APPLICATION

1. Arborescence
2. Pages et composants
3. Sécurités
4. Problématiques rencontrées
5. Exemple navigation
6. Exemple de formulaire de mise à jour du profil

CONCEPTION DE L'ESPACE ADMINISTRATEUR

1. Conception de la partie administration
2. User Story
3. Choix du langage et framework
4. Conception du frontend du site web
 1. Charte graphique
 2. Maquettage
5. Conception du back end du site web

CONCLUSION

ANNEXES