

## Rapport de stage d'été

**SPÉCIALITÉ : Génie Logiciel**

**SUJET DU RAPPORT DE STAGE :**

### **Conception et Développement d'un Système de Collecte de Feedback**

**Réalisé par: Mohamed Soltan**

**Encadrant Entreprise: Mr Maher Ben Dhaher**



# *Remerciements*

*Du profond de mon cœur,*

*Je tiens à exprimer ma sincère gratitude à l'équipe de MRC Survey pour l'opportunité qui m'a été donnée de réaliser mon stage au sein de leur startup. Je souhaite tout particulièrement remercier Mr Mohamed Turki, le directeur de la startup, pour m'avoir accepté en tant que stagiaire. Son soutien et sa confiance ont été très appréciés.*

*Je remercie également Mme Salma Fliss, le project manager, pour son accompagnement et ses conseils précieux durant cette période. Sa disponibilité et ses retours constructifs m'ont aidé à progresser dans mes missions.*

*Un grand merci à Mr Maher Ben Dhaher, mon encadreur, pour son soutien inestimable tout au long de mon stage. Ses orientations et son expertise m'ont permis de m'améliorer et d'acquérir de nouvelles compétences.*

*Je remercie également ma famille et mes amis pour leur soutien moral durant cette période.*

*Enfin, je souhaite remercier mes collègues pour leur accueil chaleureux et leur esprit d'équipe, qui ont rendu mon intégration rapide et agréable.*

***Soltan Mohamed***

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Cadre général du projet</b>	<b>2</b>
Introduction.....	3
1.1 Présentation de l'entreprise .....	3
1.1.1 Création de MRC Survey .....	3
1.1.2 Les valeurs de MRC Survey .....	4
1.2 Étude de l'existant .....	4
1.2.1 Description de l'existant .....	4
1.2.2 Problématique.....	4
1.2.3 Solution proposée .....	4
1.3 Méthodologie de travail.....	5
1.3.1 Le concept agile .....	5
1.3.2 Présentation de la méthodologie Scrum.....	5
Conclusion .....	7
<b>2 Analyse fonctionnelle</b>	<b>8</b>
Introduction.....	9
2.1 Capture des besoins.....	9
2.1.1 Identification des acteurs .....	9
2.1.2 Identification des besoins fonctionnels .....	10
2.1.3 Identification des besoins non fonctionnels .....	10
2.2 Pilotage du projet avec Scrum.....	11
2.2.1 Équipe et rôles.....	11
2.2.2 Backlog du produit.....	12
2.2.3 Planification des sprints .....	14
2.3 Solution adoptée .....	14
2.4 Les langages utilisées.....	14
2.5 Les logiciels utilisés.....	15
Conclusion .....	15

<b>3 Développement et intégration</b>	<b>16</b>
Introduction.....	17
3.1 Développement de la partie Frontend .....	17
3.1.1 Création du composant de notation.....	17
3.1.2 Gestion des événements .....	18
3.1.3 Validation du formulaire .....	19
3.1.4 Communication avec le Backend .....	20
3.2 Développement de la partie Backend.....	20
3.2.1 Création de l'API .....	20
3.2.2 Connexion à la base de données MongoDB.....	21
3.2.3 Gestion des erreurs et validations .....	21
3.3 Compilation.....	22
3.3.1 Compilation du Frontend .....	22
3.3.2 Compilation du Backend.....	22
3.4 Manipulation de la base de données .....	23
3.4.1 Ajout d'une évaluation .....	23
3.4.2 Récupération des évaluations.....	24
3.4.3 Filtrage et analyse des données .....	24
Conclusion .....	25
<b>Conclusion générale</b>	<b>26</b>
<b>Bibliographie</b>	<b>27</b>
<b>Annexes</b>	<b>28</b>
Annexe A. Fonctions de manipulation des données .....	28
Annexe B. Exemples de fonctions relatives au système « Happy or Not ».....	29
Annexe C. Génération des fichiers CSV .....	30

# Table des figures

1.1	Logo de l'entreprise MRC Survey.....	3
1.2	Vue globale « Scrum » .....	7
2.1	Diagramme de cas d'utilisation .....	10
3.1	Formulaire de rating coté Frontend.....	17
3.2	Formulaire de rating coté Frontend (survoler).....	18
3.3	Formulaire de rating coté Frontend (commenter).....	18
3.4	Formulaire de rating coté Frontend (après l'évaluation) .....	19
3.5	Formulaire de rating coté Frontend (validation de formulaire) .....	19
3.6	Formulaire de rating coté base de données (1) .....	20
3.7	Formulaire de rating coté base de données (2) .....	21
3.8	Compilation du Frontend .....	22
3.9	Compilation du Backend.....	22
3.10	Résultats d'évaluations.....	23
3.11	Récupération des évaluations CSV .....	24
3.12	Résultats d'évaluations PDF .....	24
3.13	Résumé des évaluations PDF .....	25

# Liste des tableaux

2.1	Identification des acteurs .....	9
2.2	Backlog produit .....	13

# Glossaires

—	<b>AJAX</b>	=	<b>A</b> ynchronous <b>J</b> ava <b>S</b> cript <b>A</b> nd <b>X</b> ML
—	<b>CSS</b>	=	<b>C</b> ascading <b>S</b> tyle <b>S</b> heets
—	<b>DSI</b>	=	<b>D</b> épartement du <b>S</b> ystèmes d' <b>I</b> nformation
—	<b>HTML</b>	=	<b>H</b> yper <b>T</b> ext <b>M</b> arkup <b>L</b> anguage
—	<b>IT</b>	=	<b>T</b> echnologies de l' <b>I</b> nformation
—	<b>JS</b>	=	<b>J</b> ava <b>S</b> cript
—	<b>MVC</b>	=	<b>M</b> odèle <b>V</b> ue <b>C</b> ontrôleur
—	<b>NT</b>	=	<b>N</b> ouvelle <b>T</b> echnologie
—	<b>PDF</b>	=	<b>P</b> ortable <b>D</b> ocument <b>F</b> ormat
—	<b>PL</b>	=	<b>P</b> rocedural <b>L</b> anguage
—	<b>RH</b>	=	<b>R</b> essources <b>H</b> umaines
—	<b>SASS</b>	=	<b>S</b> yntactically <b>A</b> wesome <b>S</b> tyle <b>S</b> heets
—	<b>SE</b>	=	<b>S</b> ystème d' <b>E</b> xploitation
—	<b>SGBD</b>	=	<b>S</b> ystème de <b>G</b> estion de <b>B</b> ase de <b>D</b> onnées
—	<b>SI</b>	=	<b>S</b> ystème <b>I</b> nformatique
—	<b>DB</b>	=	<b>D</b> ata <b>B</b> ase
—	<b>UC</b>	=	<b>U</b> se <b>C</b> ase
—	<b>UML</b>	=	<b>U</b> nified <b>M</b> odeling <b>L</b> anguage
—	<b>URL</b>	=	<b>U</b> niform <b>R</b> esource <b>L</b> ocator

# Introduction générale

De nos jours, la transformation numérique des entreprises devient une nécessité afin d'améliorer l'efficacité, la flexibilité et la sécurité des processus. Abandonner le papier au profit des outils numériques permet non seulement de gagner du temps, mais aussi de faciliter la gestion des données tout en assurant leur traçabilité. C'est dans cette optique que la startup MRC Survey a entrepris le développement d'une application destinée à ses clients, dans le but de remplacer les formulaires papier par une solution numérique plus performante et intuitive.

L'objectif principal de cette application est de simplifier et d'accélérer le processus de gestion des formulaires, permettant ainsi aux utilisateurs de gagner du temps et de réduire les coûts opérationnels. En plus de cela, la solution numérique permet une meilleure praticité et satisfaction pour les clients.

Pendant mon stage au sein de MRC Survey, j'ai eu pour mission de développer une section dédiée à l'évaluation de l'application, appelée "Happy or Not".

Pendant mon stage au sein de MRC Survey, j'ai eu pour mission de développer une section dédiée à l'évaluation de l'application, appelée "Happy or Not". Cette fonctionnalité, sous forme d'un système de notation (rating), permet aux utilisateurs de fournir leurs avis sur l'application elle-même. L'objectif est de recueillir ces évaluations et de les enregistrer dans une base de données afin que les administrateurs puissent analyser les retours clients et améliorer continuellement l'expérience utilisateur.

Pour mener à bien cette mission, j'ai utilisé plusieurs outils et technologies, notamment HTML, Bootstrap, Angular, TypeScript pour la partie frontend, ainsi que JavaScript, MongoDB, Node.js pour le backend et la gestion des données. Grâce à cette solution, les administrateurs pourront visualiser les évaluations des utilisateurs, analyser leurs retours et ainsi adapter l'application pour répondre au mieux aux attentes des clients.



# CadRE g n ral DU PROJET

---

## Plan

<b>Introduction . . . . .</b>	<b>3</b>
<b>1 Pr�sentation de l'entreprise. . . . .</b>	<b>3</b>
<b>2 �tude de l'existant . . . . .</b>	<b>4</b>
<b>3 M�thodologie de travail . . . . .</b>	<b>5</b>
<b>Conclusion . . . . .</b>	<b>7</b>

## Introduction

Ce chapitre est consacré à l'analyse de MRC Survey, la startup dans laquelle j'ai effectué mon stage.

### 1.1 Présentation de l'entreprise

MRC Survey a pour mission d'aider les TPE et PME à atteindre l'excellence dans leurs normes et leurs opérations. Grâce à une solution numérique simple à utiliser, une expertise fiable et un support client personnalisé, MRC Survey a su gagner la confiance de plusieurs enseignes renommées dans le secteur du Retail.



**Figure 1.1:** Logo de l'entreprise MRC Survey

#### 1.1.1 Création de MRC Survey

MRC **Survey** a été créée en réponse à un besoin croissant de digitalisation des processus depuis les années 2000. Ce besoin s'est particulièrement accentué durant la crise de la Covid-19, mettant en lumière l'importance d'adopter des solutions numériques pour faciliter la gestion quotidienne des entreprises. C'est dans ce contexte que **MRC Survey** a concrétisé son projet, en déployant plusieurs outils technologiques innovants pour répondre à ces nouveaux défis.

### 1.1.2 Les valeurs de MRC Survey

Les valeurs fondamentales de **MRC Survey** reflètent son engagement envers la qualité et l'innovation, tout en plaçant le capital humain au cœur de son activité :

- **Partage** : L'entreprise valorise le partage des connaissances et des compétences pour créer un environnement collaboratif et enrichissant.
- **Qualité** : Avec plus de 25 ans d'expérience dans le secteur du Retail, **MRC Survey** vise l'excellence à chaque point de contact client.
- **Innovation** : Convaincue du pouvoir de la digitalisation pour le progrès humain, l'entreprise mise sur des solutions technologiques innovantes.
- **Professionalisme** : L'équipe de **MRC Survey** offre des conseils personnalisés et adaptés aux attentes spécifiques de chaque client.
- **Persévérance** : L'entreprise se montre déterminée à atteindre l'excellence opérationnelle malgré les défis rencontrés.
- **Engagement** : **MRC Survey** s'engage pleinement envers le développement et le bien-être du capital humain.

## 1.2 Étude de l'existant

### 1.2.1 Description de l'existant

L'application existante de MRC Survey offre une interface intuitive pour la gestion des données, mais elle manquait d'un module simple et efficace pour recueillir les avis des utilisateurs.

### 1.2.2 Problématique

L'absence d'un mécanisme efficace de collecte des retours clients rendait difficile l'analyse rapide des données, limitant ainsi la capacité de l'entreprise à améliorer son application de manière proactive.

### 1.2.3 Solution proposée

La solution proposée, baptisée "Happy or Not", visait à fournir un outil simple et interactif permettant aux utilisateurs de noter leurs expériences. Ce module devait s'intégrer de manière fluide à l'application existante tout en offrant une analyse structurée des retours collectés.

## **1.3 Méthodologie de travail**

Une méthodologie est essentiellement un outil de planification qui aide à la mise en œuvre d'un projet, elle nous fournit un cadre et des instructions précises pour gérer efficacement notre équipe et assurer un bon déroulement des différentes phases du projet.

Pour réussir à mener à bien le processus de développement et par conséquent l'aboutissement d'un bon résultat, il est important de suivre une méthodologie adaptée.

### **1.3.1 Le concept agile**

La méthodologie Agile a été adoptée pour permettre une flexibilité dans le développement et une adaptation aux besoins émergents. Cette approche a permis de livrer des incréments fonctionnels de manière itérative.

Grâce à l'agilité, on se retrouve aussi avec un focus sur la simplicité, l'intelligence collective et également sur la communication et la collaboration concrétisées par la présence du client au cœur du projet et par l'esprit d'équipe partagé par les acteurs métiers et les développeurs.

### **1.3.2 Présentation de la méthodologie Scrum**

La méthode Scrum fait partie du groupe des méthodes agiles. Elle représente un Framework de gestion et d'organisation de projets, reconnu pour sa flexibilité, son efficacité et son empirique. L'utilisation de ce Framework vise à satisfaire au mieux les besoins du client en s'appuyant sur plusieurs points :

#### **La planification des événements :**

- **Le Sprint :**

Il s'agit d'un incrément qui consiste à développer une partie du logiciel en une période

inférieure à 30 jours. Durant cette itération l'équipe de développement se concentre sur la réalisation des fonctionnalités demandées.

- **Planification du Sprint (Sprint Planning) :**

Avant chaque sprint, une réunion de planification s'organise. Ce planning sélectionne dans le backlog du produit les exigences les plus prioritaires pour le client.

- **Mêlée quotidienne (Daily Scrum Meetings) :**

C'est une réunion quotidienne de 15 minutes qui vise à évaluer l'avancement du travail et à garder l'équipe concentrée sur l'objectif du sprint.

- **Revue du Sprint (Sprint Review) :**

À chaque fin de sprint, l'équipe présente l'incrément logiciel développé au client, celui-ci valide les différents points du backlog et indique les changements à opérer.

- **Rétrospective du Sprint (Sprint Retrospective) :**

C'est une pratique qui permet d'améliorer l'équipe et le processus de développement quotidiennement.

### **La définition des artefacts Scrum :**

- **Backlog du produit :**

Il s'agit d'un référentiel des exigences et des fonctionnalités, fourni et tenu à jour par le Product Owner.

- **Backlog du sprint :**

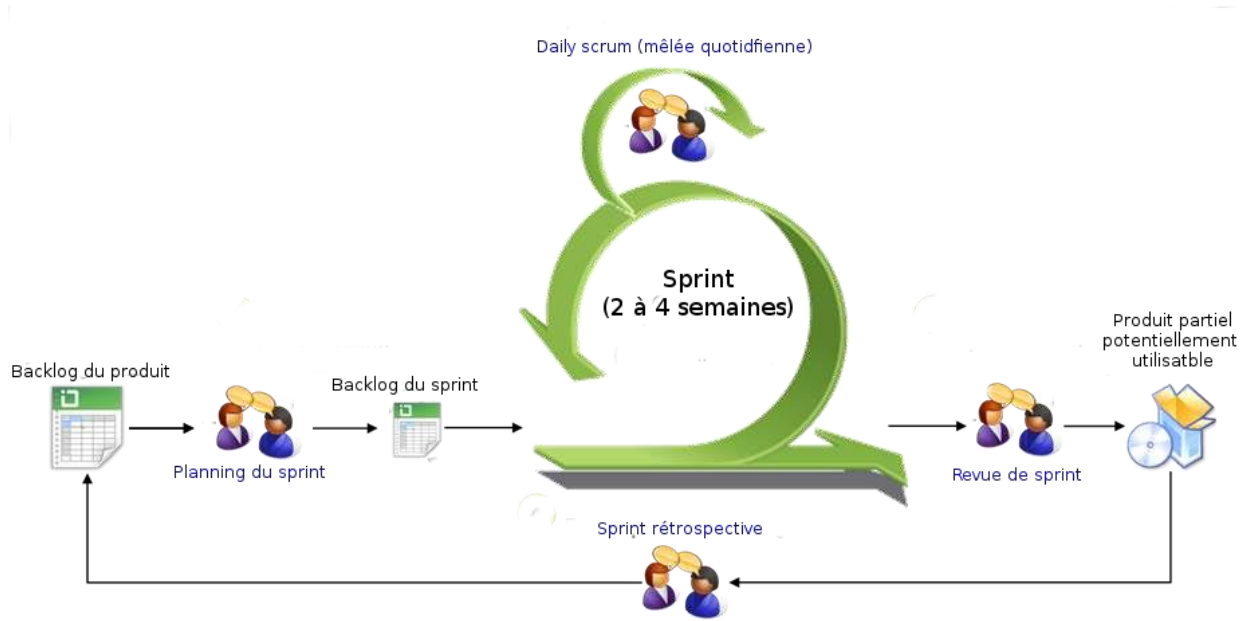
Un sous ensemble du backlog produit qui contient les fonctionnalités à mettre en oeuvre durant le sprint.

- **Incrément du produit :**

La sommation de toutes les fonctionnalités terminées pendant un sprint.

- **Le burndown chart du sprint (Sprint Burn-Down Chart) :**

Diagramme indiquant les tâches restantes avant la fin du Sprint sur un axe du temps.



**Figure 1.2 :** présente une vue globale du cycle de vie de la méthodologie « Scrum ».

## Conclusion

Ce chapitre a offert une vue d'ensemble du contexte dans lequel le projet s'inscrit.

La présentation de l'entreprise a permis de comprendre son domaine d'activité, sa mission et ses valeurs, offrant ainsi une meilleure appréhension des besoins spécifiques liés au projet. L'étude de l'existant a mis en lumière les limites et les opportunités du système actuel, justifiant la nécessité de proposer une solution adaptée. Enfin, la méthodologie de travail adoptée a été décrite, soulignant une approche structurée et collaborative pour atteindre les objectifs fixés.

Ces éléments posent les fondations solides pour la suite du projet, en assurant une cohérence entre les attentes de l'entreprise et les solutions envisagées.

# Analyse fonctionnelle

---

**Plan**

**Introduction .....9**

**1    Capture des besoins..... 9**

**2    Pilotage du projet avec Scrum..... 11**

**3    Solution adoptée ..... 14**

**4    Les langages utilisées ..... 14**

**5    Les logiciels utilisés ..... 15**

**Conclusion..... 15**

## Introduction

Au niveau de ce chapitre, nous allons aborder dans un premier temps la partie de la capture des besoins, où nous allons identifier les différents acteurs et dégager les besoins fonctionnels et non fonctionnels auxquels doit répondre notre solution. Dans un second temps, nous allons modéliser toutes les fonctionnalités par un diagramme de cas d'utilisation global. Enfin, nous allons piloter notre projet avec la méthodologie Scrum.

## 2.1 Capture des besoins

### 2.1.1 Identification des acteurs

Les principaux acteurs identifiés sont les utilisateurs finaux et les administrateurs de l'application.

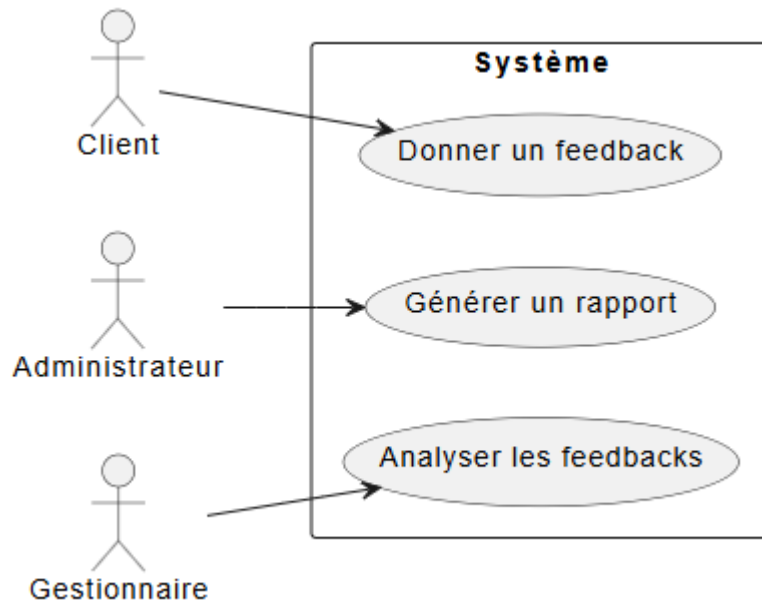
Ces acteurs peuvent être classés comme l'indique le tableau 2.1 :

Acteurs secondaires	
Système de notifications	Cet acteur est sollicité pour envoyer des alertes ou des rappels par email ou SMS aux utilisateurs.
Acteurs principaux	
Administrateur	L'administrateur supervise l'ensemble du système. Il a pour rôle de gérer les droits d'accès, d'ajouter ou de supprimer des utilisateurs, de configurer les questionnaires, et d'analyser les rapports globaux sur les évaluations.
Utilisateur final	C'est l'acteur principal qui utilise le système pour exprimer son opinion ou donner une évaluation via les dispositifs interactifs (ex : boutons de satisfaction, questionnaires). Les retours de cet utilisateur alimentent la base de données pour les analyses ultérieures.
Responsable d'évaluation	Ce responsable suit les résultats des évaluations recueillies. Il génère des rapports détaillés, propose des améliorations basées sur les retours des utilisateurs, et configure les paramètres des enquêtes (fréquence, type de questions, etc.).

**Tableau 2.1 :** Identification des acteurs

Le diagramme illustré par la figure 2.1 décrit les différents besoins fonctionnels de notre application, chaque cas d'utilisation représente une fonctionnalité offerte par le système à ses utilisateurs.





**Figure 2.1 :** Diagramme de cas d'utilisation

### 2.1.2 Identification des besoins fonctionnels

Les besoins identifiés étaient :

- Un système interactif de notation basé sur des icônes.
- Une possibilité pour les utilisateurs de laisser des commentaires complémentaires.
- Un tableau de bord pour visualiser et analyser les retours.

### 2.1.3 Identification des besoins non fonctionnels

Les besoins non fonctionnels permettent l'amélioration de la qualité logicielle de notre système. Ils agissent comme des contraintes à prendre en considération pour mettre en place une solution adéquate à ce qui est attendu, et pour éviter plusieurs incohérences dans le système. Notre application doit nécessairement répondre aux besoins suivants :

#### **Convivialité :**

L'application doit être facile à utiliser. Elle devra offrir des interfaces conviviales, simples et ergonomiques.

**Performance :**

Vu l'ampleur des données qu'on va stocker, l'application doit répondre aux besoins des utilisateurs d'une manière optimale en termes de temps de réponse, de traitement et de chargement des données.

**Maintenabilité :**

Le code doit être lisible, commenté et bien structuré afin d'obéir à l'évolution et aux différents changements des besoins.

## **2.2 Pilotage du projet avec Scrum**

### **2.2.1 Équipe et rôles**

La méthodologie Scrum fait intervenir trois rôles principaux qui sont :

- **Product owner** : C'est le représentant des clients, il définit l'ordre dans lequel les fonctionnalités seront développées.
- **Scrum Master** : Il agit comme un « Coach » et non pas comme un dirigeant, son rôle est d'aider l'équipe à avancer de manière autonome en cherchant en permanence à s'améliorer.
- **Équipe** : l'équipe est constituée des personnes qui seront chargées d'implémenter les besoins du client. Elle doit tout faire pour délivrer le produit dans les délais prévus.
- **Intervenants** : ils observent et donnent des conseils à l'équipe.

### 2.2.2 Backlog du produit

Le backlog du produit est l'artefact le plus important de Scrum, il consiste à présenter la liste des fonctionnalités ou encore les « User Stories » constituant le produit souhaité. Chaque fonctionnalité est caractérisée par un numéro ID, un thème, une priorité et une estimation de l'effort nécessaire à une équipe pour l'implémenter. Nous choisissons de représenter l'estimation sous la forme de points relatifs en utilisant la suite de Fibonacci afin d'éviter les confusions dues aux estimations trop proches et de représenter la priorité suivant la méthode

«MoSCoW» qui a pour signification :

- **M** : « Must have », toutes les fonctionnalités qui sont indispensables et nécessaires.
- **S** : « Should have », les tâches qui sont essentielles, mais pas obligatoires, elles seront développées une fois que celles de la catégorie Must ont été livrées.
- **C** : « Could have », se sont des tâches qu'ont les fait seulement s'il reste assez de temps.
- **W** : « Won't have this time but would like in the future », se sont les tâches supplémentaires, très secondaires.

Le tableau 2.2 résume le backlog du produit de notre application.

ID	Tache	ID	User stories	Priorité	Estimation
1	Collecte des avis des utilisateurs	1.1	Permettre aux utilisateurs de soumettre des évaluations de leur expérience.	Haute	5 jours
2	Visualisation des résultats	2.1	Afficher un tableau de bord avec les résultats des évaluations.	Haute	7 jours
3	Système de notification	3.1	Alerter les administrateurs lors de l'envoi d'une nouvelle évaluation.	Moyenne	3 jours
4	Gestion des comptes utilisateurs	4.1	Permettre la création, l'édition et la suppression de comptes utilisateurs.	Haute	4 jours
5	Feedback de satisfaction	5.1	Ajouter un questionnaire pour que les utilisateurs puissent donner un retour détaillé sur leur expérience.	Moyenne	6 jours
6	Historique des évaluations	6.1	Permettre aux administrateurs de visualiser les évaluations passées.	Basse	4 jours
7	Analyse des données	7.1	Intégrer un module d'analyse des résultats d'évaluation pour extraire des tendances.	Haute	8 jours
8	Rapports PDF	8.1	Générer des rapports en PDF sur les évaluations collectées.	Moyenne	5 jours

**Tableau 2.2 : Backlog produit**

### 2.2.3 Planification des sprints

La réunion de la planification des sprints est l'un des événements Scrum le plus important, puisqu'elle rythme tout le projet et sert à bien dérouler les différents sprints.

À travers cette réunion, nous évoquons le planning de notre travail qui s'est étalé sur une période de trois mois avec une estimation de deux à trois semaines pour chaque sprint.

## 2.3 Solution adoptée

Pour résoudre ces problématiques, j'ai adopté une approche basée sur l'utilisation de NoSQL (MongoDB) pour stocker les évaluations. Chaque évaluation est enregistrée avec un identifiant unique correspondant à l'utilisateur et à la session, afin de garantir l'unicité des réponses. De plus, une validation côté frontend a été mise en place pour vérifier que tous les champs requis sont correctement remplis avant soumission.

Enfin, une API basée sur Node.js a été développée pour gérer les interactions entre l'application frontend et la base de données. Cette API permet de soumettre, récupérer et filtrer les évaluations en temps réel.

## 2.4 Les langages utilisées

Pour mener à bien ce projet, plusieurs langages de programmation et outils ont été utilisés :

- **HTML5** et **CSS3** : pour la structure et le style du formulaire de notation.
- **JavaScript** et **TypeScript** : pour la gestion des interactions utilisateur, la validation des données et la communication avec l'API.
- **Angular** : pour le développement frontend, notamment la création des composants et la gestion de l'état des données.
- **Node.js** : utilisé pour développer l'API qui communique avec la base de données MongoDB.
- **MongoDB** : pour stocker les données des évaluations de manière flexible et scalable.

## 2.5 Les logiciels utilisés

Les logiciels et environnements suivants ont été utilisés pour le développement de la fonctionnalité "Happy or Not" :

- **Visual Studio Code** : comme éditeur de code principal.
- **Postman** : pour tester les requêtes API et vérifier la bonne transmission des données entre le frontend et le backend.
- **Git** : pour le versionnement du code et la collaboration avec l'équipe.
- **MongoDB Compass** : pour la visualisation et la gestion des données stockées dans la base de données.

## Conclusion

Ce chapitre a permis de poser les bases essentielles à la réalisation du projet en explorant les aspects clés de son développement. L'identification des besoins a constitué une étape cruciale, assurant que chaque fonctionnalité répond aux attentes des utilisateurs. Le choix d'une gestion agile via la méthode Scrum a offert une approche structurée et collaborative pour piloter le projet efficacement.

Par ailleurs, les solutions techniques adoptées, les langages de programmation et les outils logiciels sélectionnés ont été soigneusement analysés pour garantir la faisabilité et la performance du système. L'ensemble de ces éléments constitue une base solide pour entamer la phase de réalisation, tout en assurant une cohérence entre les objectifs et les moyens mis en œuvre.

# DEVELOPPEMENT ET INTEGRATION

---

**Plan**

**Introduction ..... 17**

**1 Développement de la partie Frontend ..... 17**

**2 Développement de la partie Backend..... 20**

**3 Compilation ..... 22**

**4 Manipulation de la base de données ..... 23**

**Conclusion..... 25**

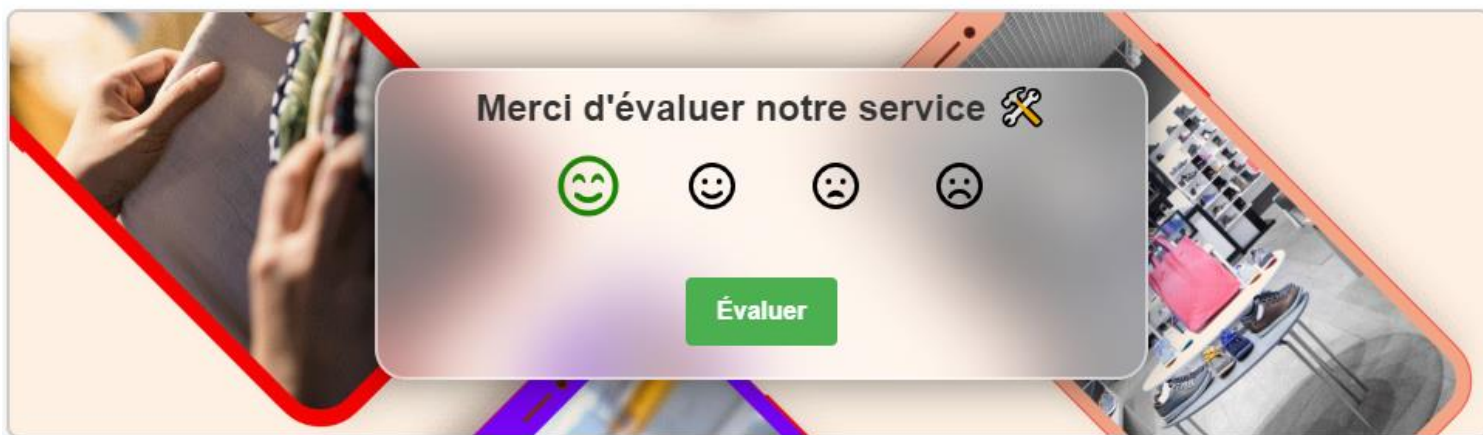
## Introduction

Ce chapitre aborde les aspects techniques liés au développement et à l'intégration de la fonctionnalité "Happy or Not" dans l'application. Le processus de développement comprend deux volets principaux : la partie **Frontend** pour l'interface utilisateur et la partie **Backend** pour la gestion des données. L'intégration de ces deux composantes est essentielle pour assurer la fluidité et l'efficacité de la fonctionnalité.

### 3.1 Développement de la partie Frontend

#### 3.1.1 Création du composant de notation

J'ai créé un composant dédié dans **Angular** pour afficher les icônes de satisfaction. Ce composant permet aux utilisateurs de sélectionner leur niveau de satisfaction à l'aide d'un simple clic.

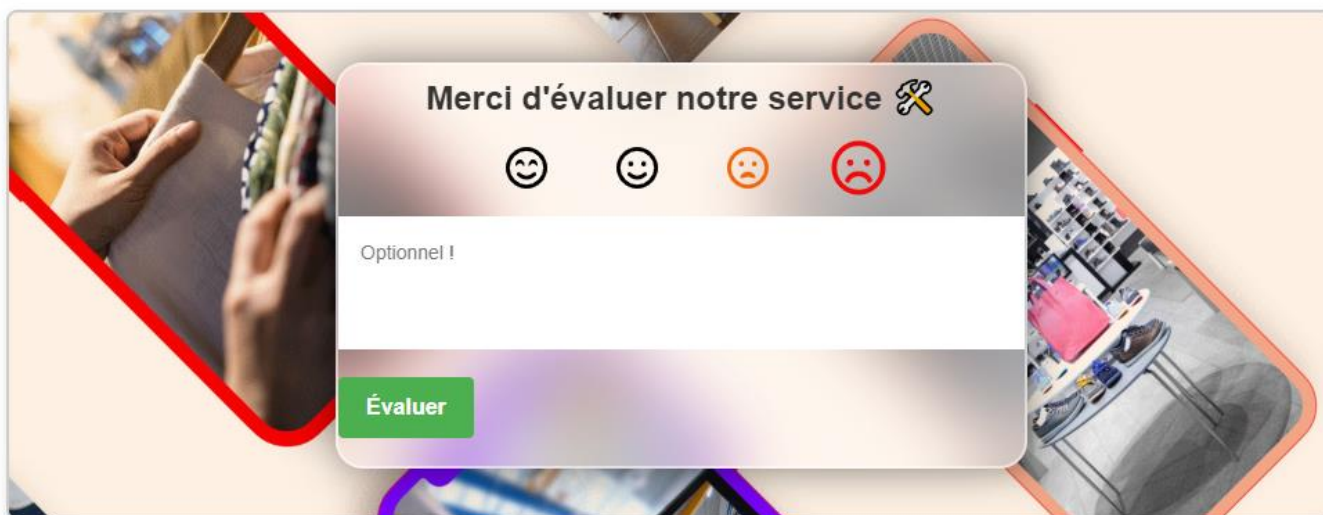


**Figure 3.1** : Formulaire de rating coté Frontend

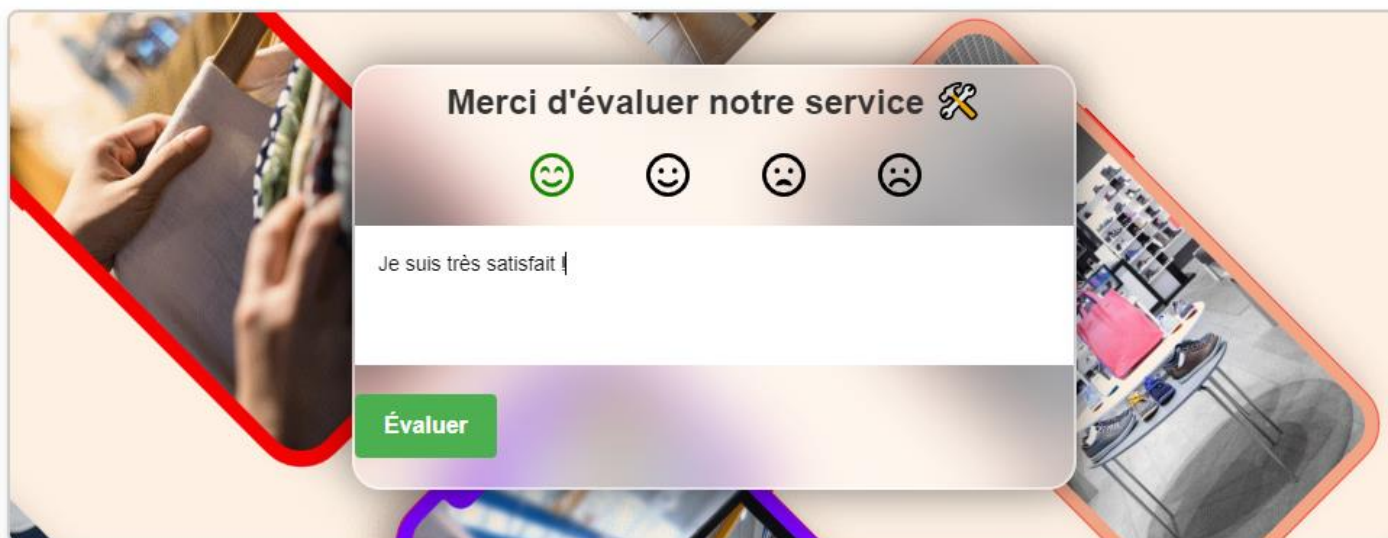


### 3.1.2 Gestion des événements

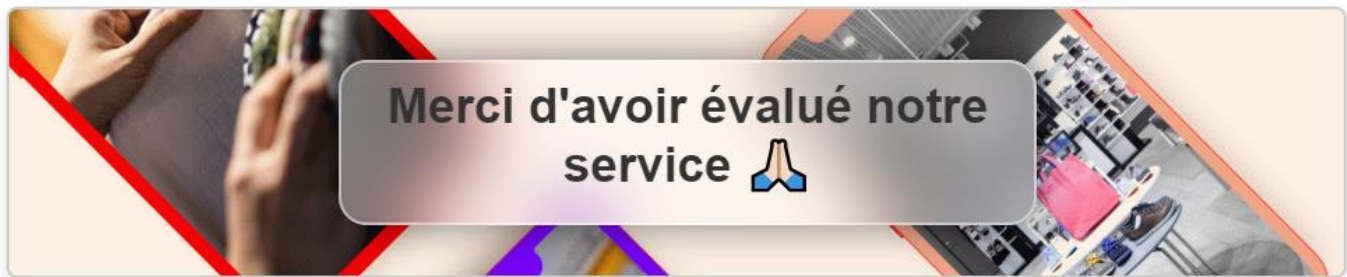
Le comportement des icônes a été géré à l'aide de **TypeScript**. Par exemple, lorsque l'utilisateur survole ou clique sur une icône, la couleur de celle-ci change pour indiquer la sélection. Cette interaction est essentielle pour offrir une expérience utilisateur fluide.



**Figure 3.2 :** Formulaire de rating coté Frontend (survoler)



**Figure 3.3 :** Formulaire de rating coté Frontend (commenter)



**Figure 3.4 :** Formulaire de rating coté Frontend (après l'évaluation)

### 3.1.3 Validation de formulaire

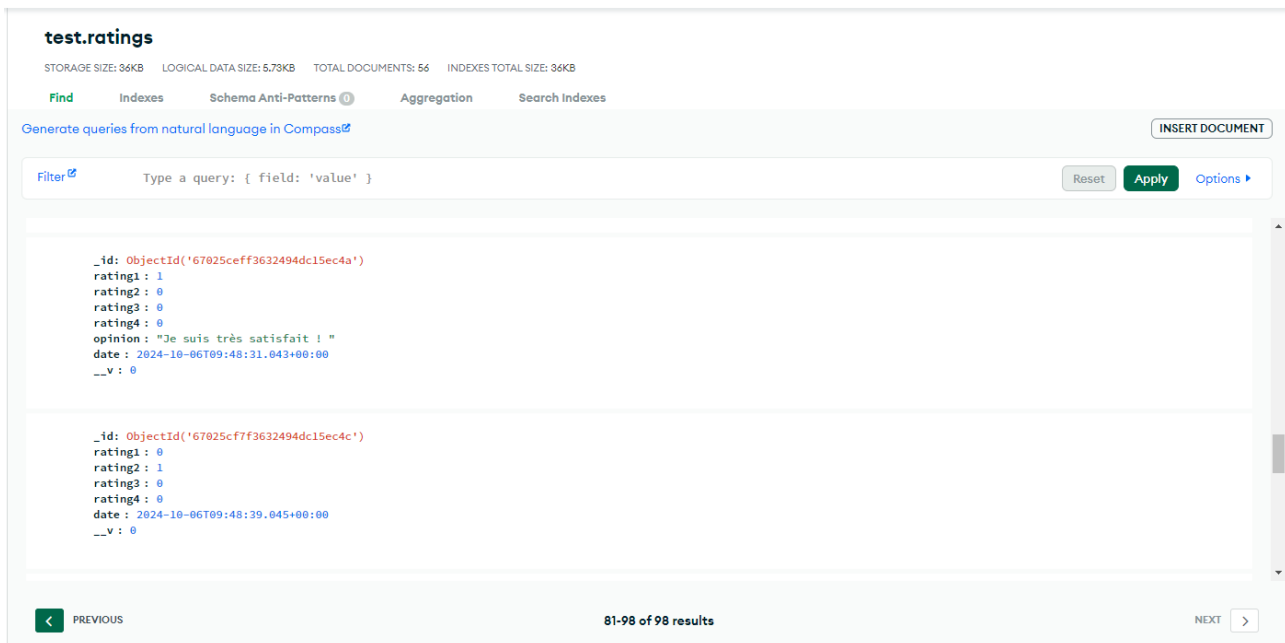
Avant de soumettre l'évaluation, une validation est effectuée pour s'assurer que toutes les conditions sont remplies (sélection d'une note, possibilité d'ajouter un commentaire). Cette validation a été implémentée en **TypeScript**.



**Figure 3.5 :** Formulaire de rating coté Frontend (Validation de formulaire)

### 3.1.4 Communication avec le Backend

Le Frontend communique avec le serveur via des requêtes **HTTP** envoyées à l'API développée en **Node.js**. Ces requêtes permettent d'envoyer les données du formulaire (notation et commentaire) vers la base de données.



**Figure 3.6 :** Formulaire de rating coté base de données (1)

## 3.2 Développement de la partie Backend

### 3.2.1 Création de l'API

Une API RESTful a été conçue pour gérer les opérations liées à la fonctionnalité "Happy or Not". Elle comprend plusieurs routes, notamment :

- POST /rating : pour soumettre une nouvelle évaluation.
- GET /ratings : pour récupérer les évaluations enregistrées.

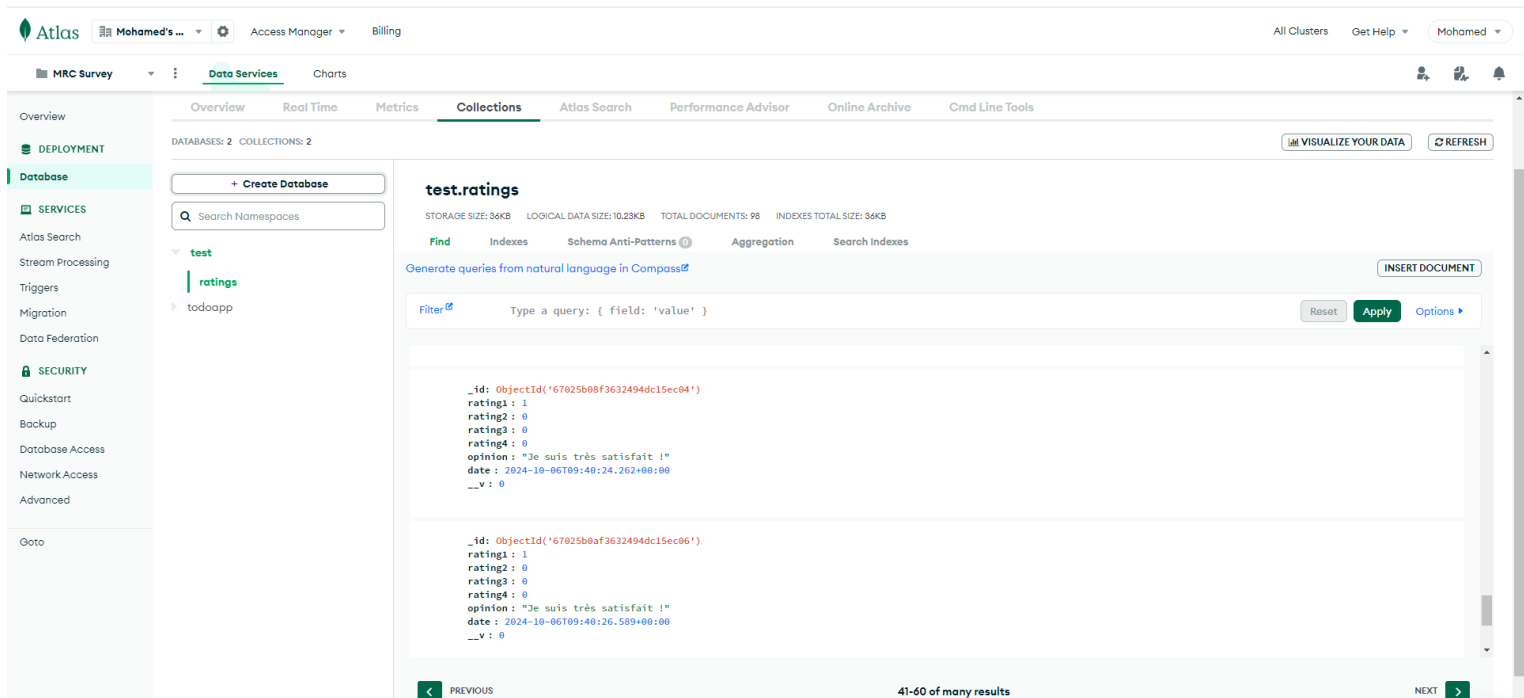
### 3.2.2 Connexion à la base de données MongoDB

J'ai utilisé **MongoDB**, une base de données NoSQL, pour stocker les évaluations des utilisateurs. Le modèle de données comprend les champs suivants :

- Niveau de satisfaction (de 1 à 4).
- Commentaire optionnel.
- Horodatage de la soumission.

### 3.2.3 Gestion des erreurs et validations

Une validation côté serveur a été mise en place pour s'assurer que les données envoyées sont conformes avant de les enregistrer dans la base de données. En cas d'erreur, des messages appropriés sont renvoyés au client pour une meilleure gestion des exceptions.

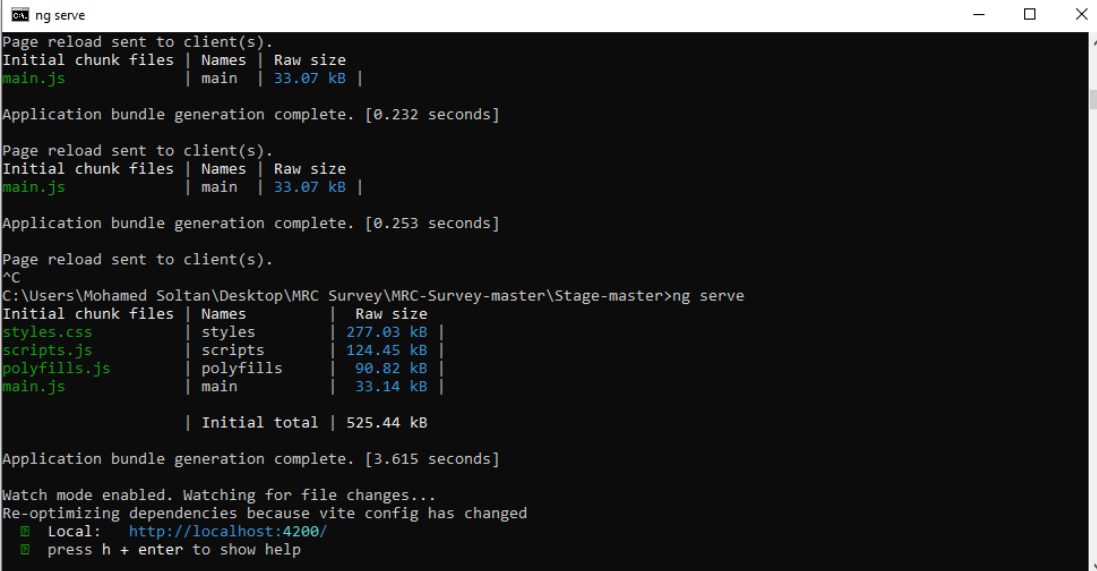


**Figure 3.7 :** Formulaire de rating coté base de données (2)

## 3.3 Compilation

### 3.3.1 Compilation de Frontend

Le code **TypeScript** et les templates **HTML** ont été transpilés en **JavaScript** et fichiers statiques prêts à être déployés sur un serveur web.

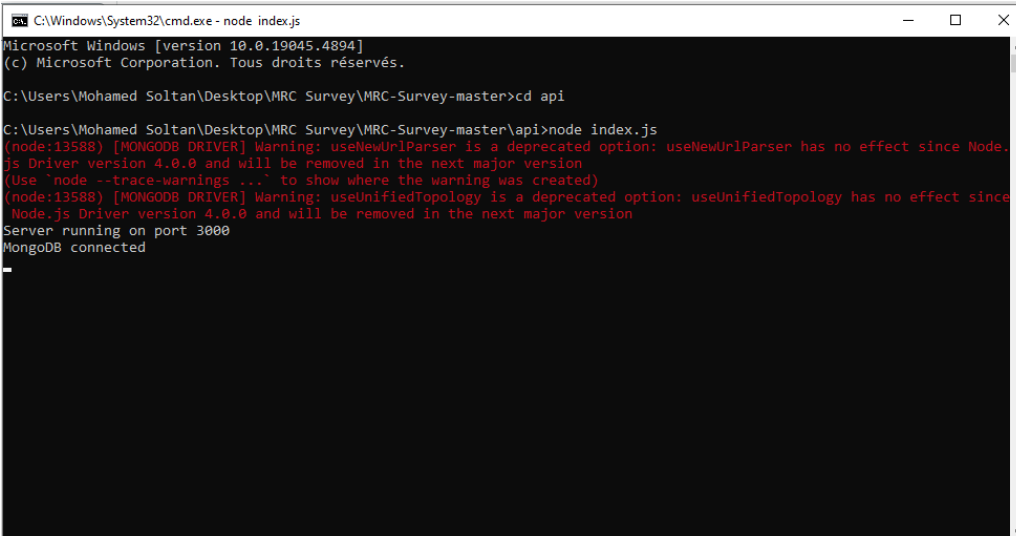


```
ng serve
Page reload sent to client(s).
Initial chunk files | Names | Raw size
main.js            | main  | 33.07 kB |
Application bundle generation complete. [0.232 seconds]
Page reload sent to client(s).
Initial chunk files | Names | Raw size
main.js            | main  | 33.07 kB |
Application bundle generation complete. [0.253 seconds]
Page reload sent to client(s).
^C
C:\Users\Mohamed Soltan\Desktop\MRC Survey\MRC-Survey-master\Stage-master>ng serve
Initial chunk files | Names | Raw size
styles.css          | styles | 277.03 kB |
scripts.js          | scripts | 124.45 kB |
polyfills.js        | polyfills | 90.82 kB |
main.js             | main  | 33.14 kB |
| Initial total | 525.44 kB
Application bundle generation complete. [3.615 seconds]
Watch mode enabled. Watching for file changes...
Re-optimizing dependencies because vite config has changed
  Local: http://localhost:4200/
  press h + enter to show help
```

Figure 3.8 : Compilation du Frontend

### 3.3.2 Compilation du Backend

Le code **Node.js** est exécuté sur le serveur, et les dépendances nécessaires sont installées à l'aide de **npm** (Node Package Manager).



```
C:\Windows\System32\cmd.exe - node index.js
Microsoft Windows [version 10.0.19045.4894]
(c) Microsoft Corporation. Tous droits réservés.

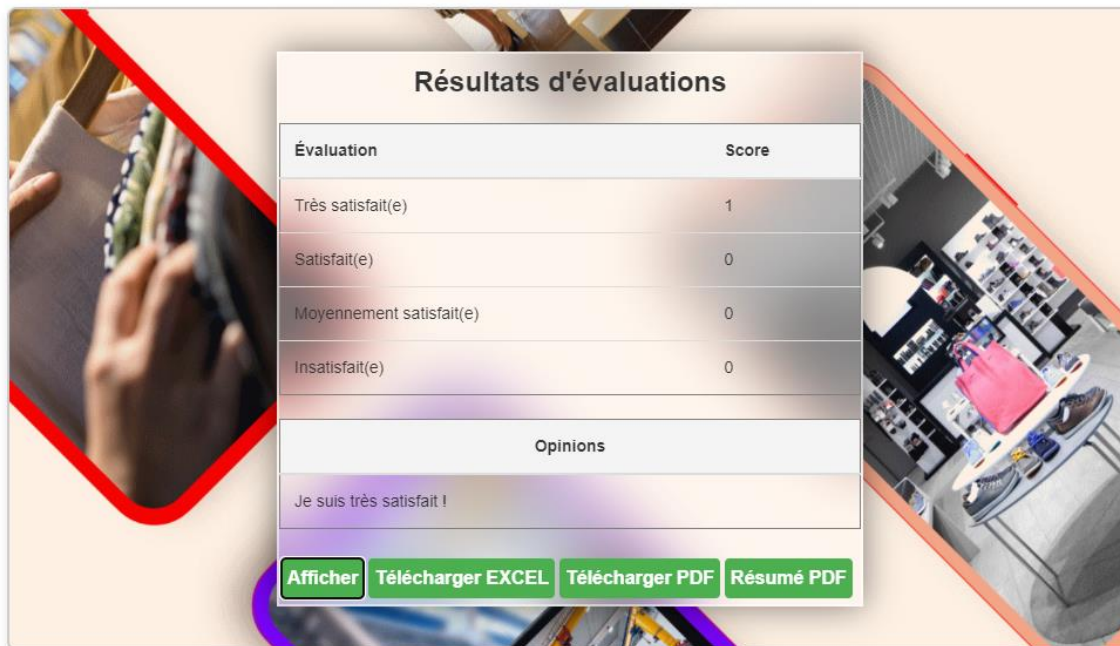
C:\Users\Mohamed Soltan\Desktop\MRC Survey\MRC-Survey-master>cd api
C:\Users\Mohamed Soltan\Desktop\MRC Survey\MRC-Survey-master\api>node index.js
(node:13588) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use 'node --trace-warnings ...' to show where the warning was created)
(node:13588) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server running on port 3000
MongoDB connected
```

Figure 3.9 : Compilation du Backend

## 3.4 Manipulation de la base de données

### 3.4.1 Ajout d'une évaluation

Chaque fois qu'un utilisateur soumet une évaluation via le formulaire, une nouvelle entrée est créée dans la collection **ratings** de la base de données. Cette entrée comprend la note, le commentaire (s'il est présent) et la date de soumission.

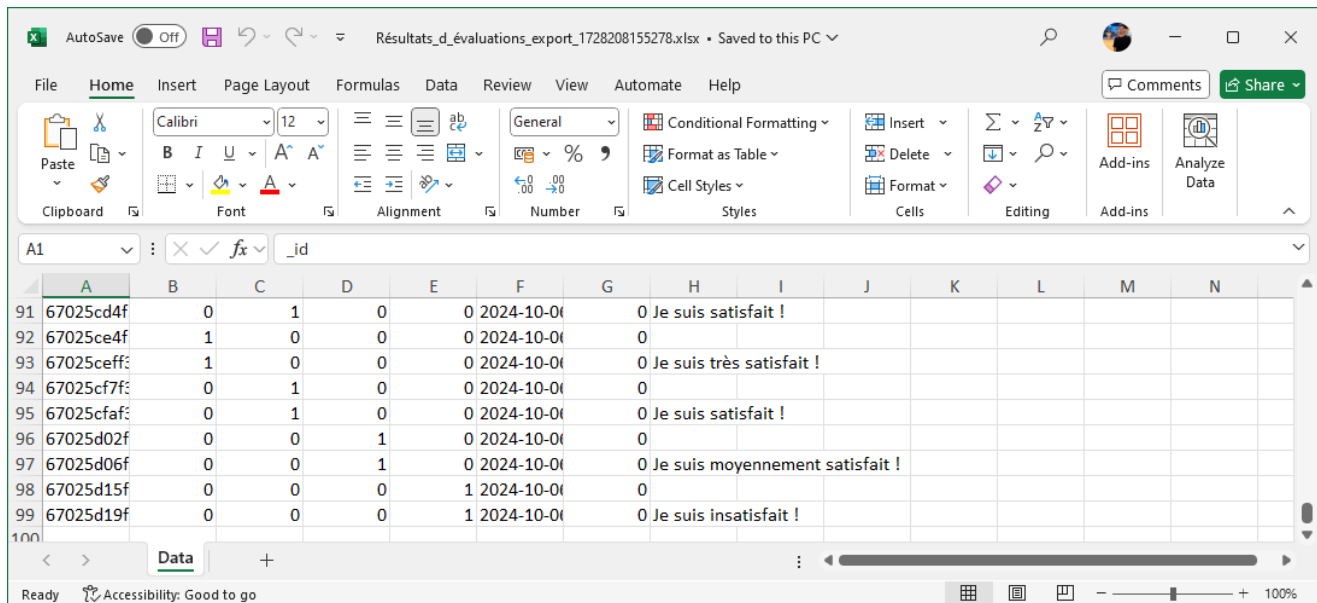


**Figure 3.10 :** Résultats d'évaluations



### 3.4.2 Récupération des évaluations

Les évaluations sont récupérées à partir de MongoDB en fonction de divers critères, tels que la date ou le niveau de satisfaction, pour permettre une analyse ultérieure par les administrateurs.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
91	67025cd4f	0	1	0	0	2024-10-01	0	Je suis satisfait !						
92	67025ce4f	1	0	0	0	2024-10-01	0							
93	67025ceff	1	0	0	0	2024-10-01	0	Je suis très satisfait !						
94	67025cf7f	0	1	0	0	2024-10-01	0							
95	67025cfaf	0	1	0	0	2024-10-01	0	Je suis satisfait !						
96	67025d02f	0	0	1	0	2024-10-01	0							
97	67025d06f	0	0	1	0	2024-10-01	0	Je suis moyennement satisfait !						
98	67025d15f	0	0	0	1	2024-10-01	0							
99	67025d19f	0	0	0	1	2024-10-01	0	Je suis insatisfait !						

Figure 3.11 : Récupération des évaluations CSV

### 3.4.3 Filtrage et analyse des données

J'ai mis en place des filtres pour afficher les évaluations par heure ou par jour, ce qui permet de comprendre les tendances d'utilisation et de satisfaction des utilisateurs sur des périodes précises.



Résultats d'évaluations	
Evaluation 1	
Très satisfait(e) :	0
Satisfait(e) :	0
Moyennement satisfait(e) :	0
Insatisfait(e) :	1
Opinions :	undefined

Figure 3.12 : Résultats d'évaluations PDF



**Figure 3.13 :** Résumé des évaluations PDF

## Conclusion

Le développement de la fonctionnalité "Happy or Not" et son intégration dans l'application ont permis de fournir un outil de feedback essentiel pour améliorer l'expérience utilisateur. Grâce à l'utilisation d'un environnement technologique moderne (Angular, Node.js, MongoDB), la solution est à la fois scalable et efficace, permettant aux administrateurs de recueillir et d'analyser les retours des utilisateurs en temps réel.



# Conclusion générale

Mon stage au sein de MRC Survey a été une expérience enrichissante, tant sur le plan technique que personnel. J'ai eu l'opportunité de participer au développement de la section "Happy or Not", une fonctionnalité essentielle pour recueillir les retours des utilisateurs sur l'application de l'entreprise. Ce projet m'a permis d'appliquer et d'approfondir mes compétences en HTML, JavaScript, Angular, Node.js et MongoDB.

J'ai appris à concevoir une interface utilisateur intuitive et à développer une API pour traiter les évaluations, tout en renforçant ma capacité à résoudre des problèmes techniques. Cette expérience m'a également permis de comprendre l'importance du feedback utilisateur dans l'amélioration continue des services offerts par MRC Survey.

Travailler dans une startup axée sur l'innovation m'a donné un aperçu précieux des dynamiques de l'entreprise et de l'impact de la transformation numérique sur les TPE et PME. Je suis reconnaissant envers l'équipe de MRC Survey pour leur soutien et leur collaboration tout au long de ce stage, qui m'a préparé à relever de futurs défis dans le domaine du développement web.

# Bibliographie

Angular Documentation. <https://angular.io/docs>

Description : Guide officiel pour le développement d'applications Angular, comprenant les concepts de base, des tutoriels avancés, et des exemples pratiques.

Node.js Documentation. <https://nodejs.org/en/docs/>

Description : Documentation officielle pour le framework Node.js, essentiel pour la gestion du backend du projet.

MongoDB Documentation. <https://www.mongodb.com/docs/>

Description : Guide détaillé sur la gestion des bases de données MongoDB, utilisé pour le stockage des évaluations.

jsPDF Library. <https://github.com/parallax/jsPDF>

Description : Librairie utilisée pour générer des fichiers PDF directement à partir des données collectées.

xlsx Library. <https://github.com/SheetJS/sheetjs>

Description : Documentation pour la manipulation des fichiers Excel, utilisée dans la génération des fichiers CSV.

W3Schools. "Introduction to Angular Services." <https://www.w3schools.com/angular/>

Description : Un tutoriel complet expliquant les services Angular et leur utilisation pour gérer les données.

Medium. "Efficient PDF Generation with jsPDF in Angular." <https://medium.com/>

Description : Article technique expliquant la création de fichiers PDF avec jsPDF.

Angular Examples. <https://github.com/angular/angular>

jsPDF Projects. <https://github.com/parallax/jsPDF>

# Annexes

```
167   this.ratingService.saveRating(ratingData).subscribe(response => {
168     |   console.log('Rating data saved successfully', response);
169     | }, error => {
170     |   console.error('Error saving rating data', error);
171     | });
172
173   const textarea = document.querySelector('.textarea') as HTMLTextAreaElement;
174   const opinion = textarea.value;
175   const opinionWithSymbol = `${opinion}`;
176   window.localStorage.setItem("opinion", opinionWithSymbol);
177
178   this.route.navigate(['/congrats']);
179 }
180 }
```

## Annexe A : Fonctions de manipulation des données

- **Enregistrement des données d'évaluation :**

Utilisation de `ratingService.saveRating()` pour sauvegarder les données sur le serveur. Les réponses réussies et les erreurs sont traitées dans des callbacks distincts.

- **Sauvegarde de l'opinion localement :**

Le contenu de l'élément `textarea` est récupéré, transformé si nécessaire, puis sauvegardé dans `localStorage` pour un usage ultérieur.

- **Redirection :**

Une fois les données sauvegardées, l'utilisateur est redirigé vers une page de félicitations pour améliorer l'expérience utilisateur.

```

105 savePDF() {
106   this.ratingService.getRatings().subscribe(ratings => {
107     const doc = new jsPDF();
108     let yPosition = 20;
109     const pageHeight = doc.internal.pageSize.height;
110
111     doc.setFontSize(12);
112     doc.text('Résultats d\'évaluations', 15, yPosition);
113     yPosition += 10;
114
115     ratings.forEach((rating, index) => {
116       if (yPosition + 30 > pageHeight) {
117         doc.addPage();
118         yPosition = 20;
119       }
120
121       doc.setFontSize(10);
122       doc.text(`Evaluation ${index + 1}`, 15, yPosition);
123       yPosition += 10;
124
125       const data = [
126         { "Catégorie": "Très satisfait(e)", "Valeur": rating.rating1 },
127         { "Catégorie": "Satisfait(e)", "Valeur": rating.rating2 },
128         { "Catégorie": "Moyennement satisfait(e)", "Valeur": rating.rating3 },
129         { "Catégorie": "Insatisfait(e)", "Valeur": rating.rating4 },
130         { "Catégorie": "Opinions", "Valeur": rating.opinion }
131       ];
132
133       data.forEach((item) => {
134         if (yPosition + 10 > pageHeight) {
135           doc.addPage();
136           yPosition = 20;
137         }
138
139         doc.text(`${item["Catégorie"]} : ${item["Valeur"]}`, 15, yPosition);
140         yPosition += item["Catégorie"] === "Opinions" ? 20 : 10;
141       });
142
143       yPosition += 10;
144     });
145     doc.save('Résultats_d_évaluations.pdf');
146   }, error => {
147     console.error('Error fetching ratings:', error);
148   });
149 }

```

## Annexe B : Exemples de fonctions relatives au système « Happy or Not »

- **Récupération des évaluations :**

La méthode utilise un service (ratingService) pour récupérer les données d'évaluation via une requête API.

- **Création du PDF :**

- Le fichier PDF est généré avec la bibliothèque jsPDF.
- Le titre "Résultats d'évaluations" est ajouté en haut de chaque page.
- Les évaluations sont ensuite listées avec leurs catégories et valeurs associées.

- **Gestion des dépassements de page :**

Lorsque les données dépassent la hauteur de la page, une nouvelle page est ajoutée.

- **Téléchargement :**

Une fois toutes les données insérées, le fichier PDF est téléchargé avec un nom explicite.

```
83     fileDownload() {
84         this.ratingService.getRatings().subscribe(ratings => {
85
86
87             const worksheet: XLSX.WorkSheet = XLSX.utils.json_to_sheet(ratings);
88
89             const workbook: XLSX.WorkBook = {
90                 Sheets: { 'Data': worksheet },
91                 SheetNames: ['Data']
92             };
93
94             const excelBuffer: any = XLSX.write(workbook, { bookType: 'xlsx', type: 'array' });
95
96             this.saveAsExcelFile(excelBuffer, "Résultats_d'évaluations");
97         });
98     }
99     private saveAsExcelFile(buffer: any, fileName: string): void {
100         const data: Blob = new Blob([buffer], { type: EXCEL_TYPE });
101         saveAs(data, `${fileName}_export_${new Date().getTime()}.xlsx`);
102     }
```

## Annexe C : Génération des fichiers CSV

- **Accessibilité :** Les utilisateurs peuvent exporter facilement les données d'évaluation au format CSV pour les consulter hors ligne.
- **Clarté :** Les données sont organisées en colonnes, avec un onglet intitulé Data.