Projet E5 : Système de Gestion de Liste de Tâches (TaskMastery)

1. Introduction

1.1. Contexte et justification :

Dans le monde professionnel moderne, la collaboration efficace et la gestion des tâches sont des éléments clés pour le succès de toute organisation. Avec la multiplication des équipes à distance et des projets transversaux, il est essentiel de disposer d'outils numériques qui facilitent la coordination, la communication et le suivi des tâches.
De nombreuses entreprises utilisent une variété d'outils séparés pour la gestion des tâches, la communication et le partage d'informations, ce qui peut entraîner une fragmentation des informations et une perte d'efficacité. En outre, le besoin croissant de réactivité et de flexibilité nécessite des solutions intégrées permettant une collaboration en temps réel. Et pas seulement pour les professionnels, les étudiants peuvent utiliser un gestionnaire de tâche pour les travaux en groupe.

1.2. Objectifs du projet :

Le projet de création d'un gestionnaire de tâches collaboratif avec une fonctionnalité de chat intégré répond à plusieurs besoins identifiés :

1. **Centralisation de la gestion des tâches et de la communication** : En combinant la gestion des tâches et la communication en un seul outil, ce projet vise à réduire la fragmentation des informations et à améliorer la fluidité des échanges entre les membres de l'équipe.
2. **Amélioration de la productivité** : En permettant aux utilisateurs de suivre les tâches, de partager des documents et de discuter en temps réel au sein de la même plateforme, le projet vise à réduire le temps passé à jongler entre différents outils et à augmenter la productivité globale.
3. **Facilitation de la collaboration à distance** : Avec la montée en puissance du télétravail, il est crucial de disposer d'outils permettant une collaboration aussi efficace qu'en présentiel. Ce gestionnaire de tâches avec chat intégré permet de maintenir une communication constante et de coordonner les efforts des équipes dispersées géographiquement.
4. **Réduction des délais de réponse** : La fonctionnalité de chat en temps réel permet de répondre rapidement aux questions et de résoudre les problèmes sans délai, ce qui est essentiel pour le respect des délais et la gestion efficace des projets.
5. **Amélioration de la traçabilité et de la transparence** : En centralisant les discussions et les tâches, chaque membre de l'équipe a une vue claire sur l'avancement du projet et les responsabilités de chacun, ce qui améliore la transparence et la responsabilisation.

Ce projet vise donc à répondre à un besoin croissant de solutions intégrées pour la gestion des tâches et la communication, en s'appuyant sur les meilleures pratiques de collaboration numérique.

2. Description du projet

2.1. Présentation générale :

Le projet consiste à créer un système de gestion de liste de tâches destiné à aider les utilisateurs à organiser, suivre et gérer leurs tâches de manière simple et efficace. Ce système comprendra deux interfaces : un client léger accessible depuis un navigateur web pour une utilisation rapide et intuitive, et un client lourd permettant une gestion plus approfondie avec des fonctionnalités avancées telles que le suivi de projets et des tâches.

2.1.1. Objectifs principaux

- **Faciliter la gestion des tâches quotidiennes :** Offrir une interface intuitive pour créer, modifier, et supprimer des tâches.
- **Améliorer la productivité des utilisateurs** : Fournir des outils pour prioriser les tâches, fixer des échéances et attribuer des tâches à différents membres d'une équipe.
- **Fournir des statistiques et des rapports** : Permettre aux utilisateurs d'analyser leur productivité et l'avancement des projets à travers des tableaux de bord et des rapports détaillés.

2.1.2. Périmètre du projet

- **Client léger (Web)** :
    - Accès via un navigateur web standard.
    - Fonctionnalités de base de gestion des tâches : création, modification, suppression.
    - Interface utilisateur intuitive et réactive.
- **Client lourd (Desktop)** :
    - Installation sur le système d'exploitation Windows.
    - **Fonctionnalités avancées de gestion de projet** : création de projets, assignation de tâches, suivi de l'avancement.
    - Statistiques détaillées
    - Ajouter, supprimer ou modifier les étiquettes
    - Connaître les membres de la tâche.

2.1.3. Fonctionnalités principales

- **Gestion des projets :**
    - Créer, modifier et supprimer les projets.
    - Déplacer chaque tâche avec le bon statut.
    - Accéder aux différentes tâches du projet.
- **Gestion des tâches :**
    - Créer, modifier, et supprimer des tâches.
    - Assigner des tâches à des utilisateurs.
    - Prioriser les tâches et définir des échéances.
- **Collaboration** :
    - Fonctionnalité de chat en temps réel intégrée pour la communication entre les membres de l'équipe.
    - Partage de tâches avec d'autres utilisateurs.

- **Suivi des projets** :
    - Tableaux avec toutes les tâches avec les nombre de checklists terminé.
    - Consulter les membres d’une tâche.
    - Ajouter ou supprimer ou modifier les étiquettes.

3. Spécifications fonctionnelles

3.1. Gestion des utilisateurs

- **Création de comptes utilisateurs :** Les utilisateurs doivent pouvoir créer un compte en fournissant des informations de base telles que le nom, l'adresse e-mail et un mot de passe.
- **Connexion/Déconnexion :** Les utilisateurs doivent pouvoir se connecter et se déconnecter de leur compte de manière sécurisée.
- **Gestion des profils :** Les utilisateurs doivent pouvoir mettre à jour leurs informations personnelles et préférences.

3.2. Gestion des tâches

- **Création de tâches** : Les utilisateurs doivent pouvoir créer de nouvelles tâches en fournissant un titre, une description, une date d'échéance et une priorité (étiquette).
- **Modification de tâches** : Les utilisateurs doivent pouvoir modifier les détails des tâches existantes.
- **Suppression de tâches** : Les utilisateurs doivent pouvoir supprimer des tâches qu'ils ont créées.
- **Assignation de tâches** : Les utilisateurs doivent pouvoir assigner des tâches à d'autres membres de l'équipe.
- **Dépendance des tâche**s : Les utilisateurs doivent pouvoir définir une dépendance de la tâche.

3.3. Gestion des projets

- **Création de projets** : Les utilisateurs doivent pouvoir créer des projets pour regrouper plusieurs tâches.
- **Modification de projets** : Les utilisateurs doivent pouvoir modifier les détails des projets existants.
- **Suppression de projets** : Les utilisateurs doivent pouvoir supprimer des projets qu'ils ont créés.
- **Suivi de l'avancemen**t : Les utilisateurs doivent pouvoir suivre l'avancement global d'un projet en fonction des tâches accomplies.

3.4. Collaboration et communication

- **Chat intégré** : Les utilisateurs doivent pouvoir envoyer des messages instantanés à d'autres membres de l'équipe directement depuis l'application.

3.5. Gestion des accès et des permissions

- **Rôles et permissions** : Chaque participant à le droit de modifier ou quitter une tâche à laquelle il a été invité mais pas pouvoir la supprimer.
- **Contrôle d'accès** : Les utilisateurs doivent pouvoir contrôler qui peut voir et modifier leurs tâches et projets.

3.8. Accessibilité et compatibilité

- **Accessibilité Web** : Le client léger doit être compatible avec les navigateurs web modernes et accessible sur différents appareils (ordinateurs, tablettes, smartphones).
- **Application Desktop** : Le client lourd doit être compatible avec les principaux systèmes d'exploitation.

3.9. Sécurité

- **Authentification sécurisée** : Utilisation de protocoles sécurisés pour l'authentification des utilisateurs.
- **Chiffrement des mots de passe**

4. Spécifications techniques

4.1. Client léger

4.1.1. Prérequis

- Serveur Web : Apache (inclus dans XAMPP) ou Nginx.
- Langage de programmation : PHP 7.4+.
- Framework : Laravel 8+.
- Base de données : MySQL 5.7+.
- Contrôle de version : Git.
- Éditeur de texte : Visual Studio Code.
- Environnement de développement : XAMPP pour le serveur web local.
- Gestion des dépendances : Composer (pour PHP) et npm (pour JavaScript).

4.1.2. Conception et Architecture

- Architecture MVC (Modèle-Vue-Contrôleur) :
    - Modèle : Représente la structure de la base de données et les opérations CRUD (Create, Read, Update, Delete).
    - Vue : Interface utilisateur construite avec Blade (le moteur de templates de Laravel) et Bootstrap pour le CSS.
    - Contrôleur : Gère la logique d'application, les requêtes HTTP et les interactions entre le modèle et la vue.
- Framework Laravel : Utilisé pour structurer l'application web et gérer les routages, les middlewares et l'injection de dépendances.

4.1.3. Gestion de Base de Données

- SQL : Utilisation de MySQL pour concevoir et interroger la base de données.
- MySQL Workbench : Outil pour la modélisation (MLD, MCD) et la gestion de la base de données.
- Migrations et Seeders : Utilisation des migrations Laravel pour gérer les schémas de base de données et des seeders pour générer des données de test.

4.1.4. Contrôle de Version

- Git : Utilisé pour le versionnement du code, permettant un suivi efficace des modifications.
- GitHub : Stockage du code et collaboration, permettant de travailler sur plusieurs machines et avec plusieurs contributeurs.

4.1.5. Environnement de Développement

- Visual Studio Code : L'éditeur de texte utilisé pour l'écriture du code.
- XAMPP : Utilisé pour lancer un serveur web local et tester le code PHP.
- MySQL Workbench : Utilisé pour la modélisation et la gestion de la base de données.

4.1.6. Front-End

- Figma : Utilisé pour créer des maquettes et des templates avant le développement.
- Bootstrap : Framework CSS utilisé pour créer une interface utilisateur réactive et attrayante.
- jQuery : Bibliothèque JavaScript utilisée pour simplifier la manipulation du DOM et améliorer l'interactivité.
- AJAX : Utilisé pour effectuer des requêtes asynchrones, améliorant la réactivité et l'expérience utilisateur.

4.1.7. Gestion des Dépendances

- npm : Utilisé pour gérer les dépendances CSS et JavaScript, facilitant l'intégration et la mise à jour des bibliothèques externes.
- Composer : Utilisé pour gérer les dépendances PHP.

4.1.8. Gestion des Sessions et Cookies

- Sessions et Cookies : Utilisés pour la gestion des sessions utilisateurs et l'authentification, assurant une expérience utilisateur personnalisée et sécurisée.

4.2. Client lourd

4.2.1. Prérequis

- Environnement de développement : .NET Framework 4.7.2+.
- Langage de programmation : C#.
- Framework : WPF (Windows Presentation Foundation).
- Base de données : MySQL 5.7+.
- Contrôle de version : Git.

- Éditeur de texte : Visual Studio.

4.2.2. Conception et Architecture

- Architecture MVVM (Modèle-Vue-VueModèle) :
    - Modèle : Gestion des données et de la logique métier.
    - Vue : Interface utilisateur construite avec XAML.
    - VueModèle : Liaison entre le modèle et la vue, facilitant la gestion de l'état de l'interface utilisateur et les interactions.
- Framework WPF : Utilisé pour créer des applications de bureau modernes et ergonomiques.

4.2.3. Gestion de Base de Données

- SQL : Utilisation de MySQL pour concevoir et interroger la base de données.
- MySQL Workbench : Outil pour la modélisation (MLD, MCD) et la gestion de la base de données.
- Entity Framework : Utilisé pour l'accès aux données et l'ORM (Object-Relational Mapping).

4.2.4. Contrôle de Version

- Git : Utilisé pour le versionnement du code, permettant un suivi efficace des modifications.
- GitHub : Stockage du code et collaboration, permettant de travailler sur plusieurs machines et avec plusieurs contributeurs.

4.2.5. Environnement de Développement

- Visual Studio : Environnement de développement intégré (IDE) utilisé pour l'écriture et le débogage du code.
- XAMPP : Utilisé pour tester la base de données localement.
- MySQL Workbench : Utilisé pour la modélisation et la gestion de la base de données.

4.2.6. Front-End

- Figma : Utilisé pour créer des maquettes et des templates avant le développement.
- WPF : Utilisé pour créer une interface utilisateur moderne et ergonomique en XAML

5. Suivi de l’avancement du projet

Le suivi est fait sur Trello qui me permet d’avoir une idée globale de l’avancement du projet avec beaucoup plus de détails.

6. Référence

- monday
- trello
- notion

7. Base de données