

Objectif du projet

Créer une **application web d'agenda virtuel** permettant :

- d'organiser ses rendez-vous, tâches et événements,
 - de **partager du contenu (texte, fichiers, images, liens, notes)** avec d'autres utilisateurs,
 - avec des **paramètres de visibilité** (public / privé / restreint à certaines personnes),
 - et éventuellement un **espace collaboratif** (chat, commentaires, notifications).
-

◆ Étape 1 – Définir les fonctionnalités principales

1.1. Gestion de compte utilisateur




- Inscription / connexion (email, mot de passe, ou via Google)
- Profil utilisateur (photo, nom, bio courte)
- Gestion des permissions (privé, amis, public)

1.2. Agenda personnel

- Ajouter / modifier / supprimer un événement
- Vue calendrier (jour, semaine, mois)
- Ajouter des tags ou catégories (travail, perso, études, etc.)
- Notifications / rappels

1.3. Partage de contenu

- Possibilité d'ajouter :
 - une **note textuelle**
 - un **fichier joint** (photo, doc)

- un **lien web**
- Chaque contenu a une **visibilité** :
 -  Public (visible par tous)
 -  Restreint (visible par amis / groupe)
 -  Privé (visible uniquement par soi)

1.4. Interaction sociale

- Ajouter des amis / suivre des utilisateurs
- Partager un événement avec un ou plusieurs amis
- Commentaires sur un événement partagé
- Notifications (quelqu'un t'a partagé un contenu)

1.5. (Optionnel) Fonctions avancées

- Système d'étiquettes intelligentes (IA : détecter le type d'événement)
- Synchronisation avec Google Calendar / Outlook
- Mode "collaboration de groupe" (agenda commun)
- Application mobile (avec Ionic ou React Native plus tard)

◆ Étape 2 – Conception technique

2.1. Stack

- **Frontend** : Angular
- **Backend** : Node.js + Express
- **Base de données** : MongoDB (NoSQL, facile à relier à Node.js)
- **Auth** : JWT (JSON Web Token)

- **Stockage fichiers** : Cloudinary ou Firebase Storage
- **Notifications** : Socket.io (temps réel)

2.2. Architecture

```
/frontend (Angular)
├── src/app
│   ├── components
│   ├── pages
│   ├── services
│   └── models
/backend (Node.js)
├── controllers
├── routes
├── models
├── middleware
└── utils
```

◆ Étape 3 – Plan de développement (par phase)

🧩 Phase 1 : Base du projet

- Mettre en place le serveur Node.js (Express)
- Connecter MongoDB
- Créer le système d'authentification (inscription, connexion)
- Créer la base du frontend Angular (page login/register)

Phase 2 : Agenda et gestion des événements

- Backend : CRUD pour les événements
- Frontend : vue calendrier + formulaire d'ajout
- Gestion des rappels et notifications locales

Phase 3 : Partage et visibilité

- Créer les options de partage (public / amis / privé)
- Mettre en place un système d'amis
- Partager un événement / contenu

Phase 4 : Contenu enrichi

- Ajouter upload de fichiers / images
- Lier fichiers à un événement
- Interface de visualisation (mini feed de contenus)

Phase 5 : Temps réel et notifications

- Intégrer **Socket.io** pour notifications instantanées
- Système de commentaires / réactions

Phase 6 : Optimisation et design

- Refonte du design (Angular Material, TailwindCSS)
- Amélioration de la performance et de l'UX

◆ **Étape 4 – Sécurité et performance**

- Protection des routes avec middleware JWT
- Validation des données (Joi / Zod)
- Gestion des rôles et permissions
- Caching (Redis optionnel)

◆ **Étape 5 – Hébergement**

- Backend : Render / Railway / VPS
- Frontend : Netlify / Vercel
- Base de données : MongoDB Atlas
- Stockage : Cloudinary

Souhaites-tu que je t'aide à **développer la première phase (mise en place backend + frontend de base)** avec le code et la structure du projet ?

(Je peux te faire un plan de fichiers et les premiers scripts Node.js + Angular.)