DOSSIER DE PRÉSENTATION

NEXARIS

Un espace pour relier les tâches, les idées et les décisions.

Application web de gestion de projets et d’organisation, pensée comme un Second Brain.

Statut déclaré : Terminé

Développement personnel majoritaire, avec une assistance IA occasionnelle.

Objectif

Rassembler le suivi quotidien, la planification et la documentation pour garder le contexte d’un projet à portée de main.

Utilisateurs concernés

Personnes et équipes souhaitant organiser leurs projets dans un même espace de travail.

Portée de ce dossier

Sources locales examinées. Tableau, fiche de tâche et calendrier observés en mode local, en lecture seule, avec les exemples inclus dans le code. Le site public demande une connexion ; les écritures et la synchronisation cloud n’ont pas été testées.

01 / USAGES

Fonctionnalités examinées

Suivre le travail

Tableau Kanban, priorités, étiquettes, multi-assignation et sous-tâches. Le tableau et la consultation des tâches ont été observés en mode local.

Planifier

Calendrier accessible depuis la navigation. Le code comprend aussi des jalons et une vue roadmap ; leur édition n’a pas été testée.

Déplacer et retrouver

Le code implémente le déplacement des cartes avec dnd-kit et une navigation par fragment d’URL vers les vues et les tâches. Aucun déplacement n’a été effectué pendant cette inspection.

Conserver le contexte

Modules de documentation, inspirations et réunions présents dans les sources. Les réunions comportent des décisions et une action de création de tâche ; ces écritures n’ont pas été exécutées.

Couche	Technologies identifiées
Interface	React 18, TypeScript, Tailwind CSS, Vite
État et interactions	Zustand, dnd-kit
Synchronisation optionnelle	Supabase, schéma PostgreSQL / JSONB

02 / TECHNIQUE

Deux modes de persistance

Interface React

Vues, composants de tâches et routeur par fragment d'URL.

État Zustand

Données du tableau et actions métier ; persistance locale via le middleware persist.

Stockage local

localStorage dans le navigateur. Mode disponible sans configuration Supabase.

Mode cloud optionnel

useCloudSync relie l'état à Supabase : table boards, données JSONB, contrôle updated_at et événements temps réel.

L'interface alimente le store. Le store persiste localement ; lorsque Supabase est configuré, le hook synchronise aussi le tableau distant. Le schéma SQL décrit PostgreSQL ; son état réel en production n'a pas été contrôlé.

Une tâche, plusieurs points d'entrée

Le routeur lit la vue et l'identifiant de tâche dans l'URL. Il vérifie que la tâche existe dans le tableau chargé avant d'ouvrir sa fiche. La navigation arrière peut refermer cette fiche.

Réconcilier des modifications

mergeBoards compare une base commune, l'état local et l'état distant, collection par collection. Les changements compatibles sont fusionnés ; une collision sur la même entité déclenche la gestion de conflit.

03 / INTERFACES

Le tableau et les tâches

Captures authentiques du mode local. Les données d'exemple sont déjà incluses dans les sources. L'interface porte le nom Nova. Aucun résultat de production n'est illustré.

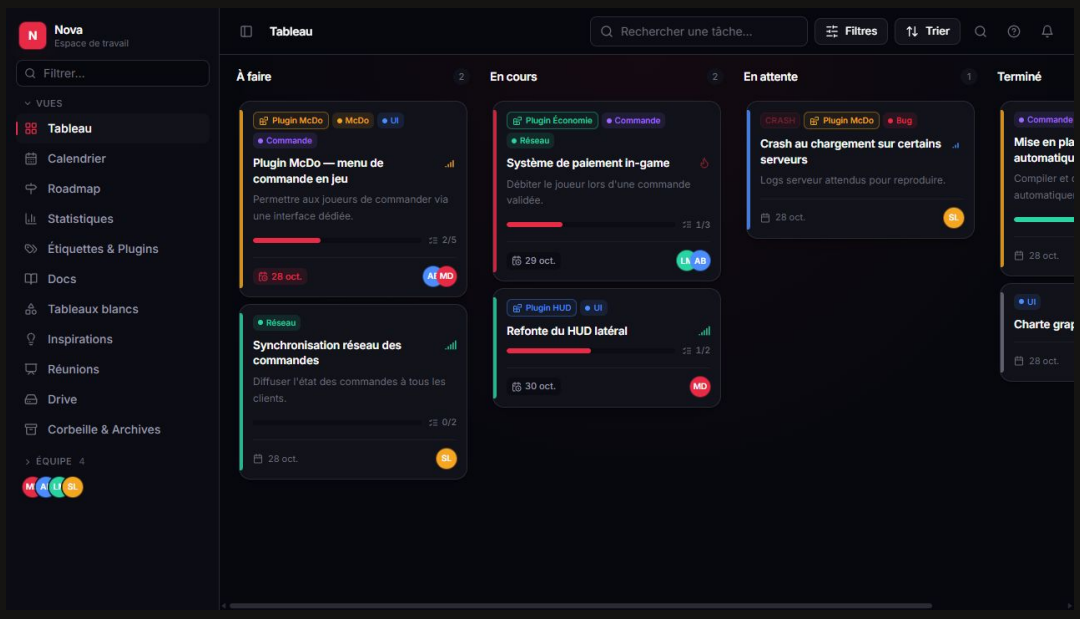
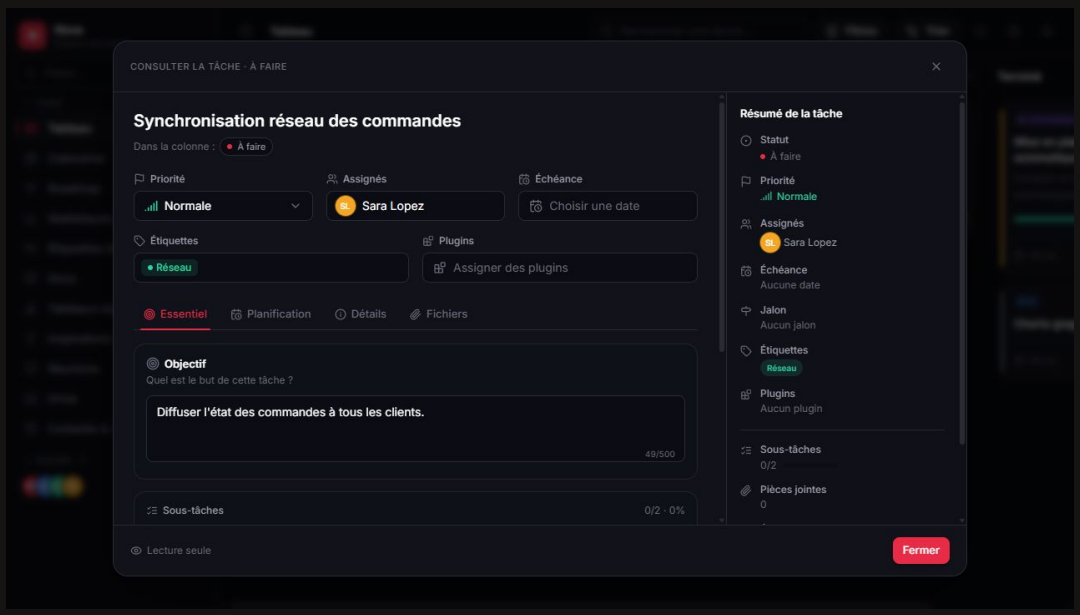


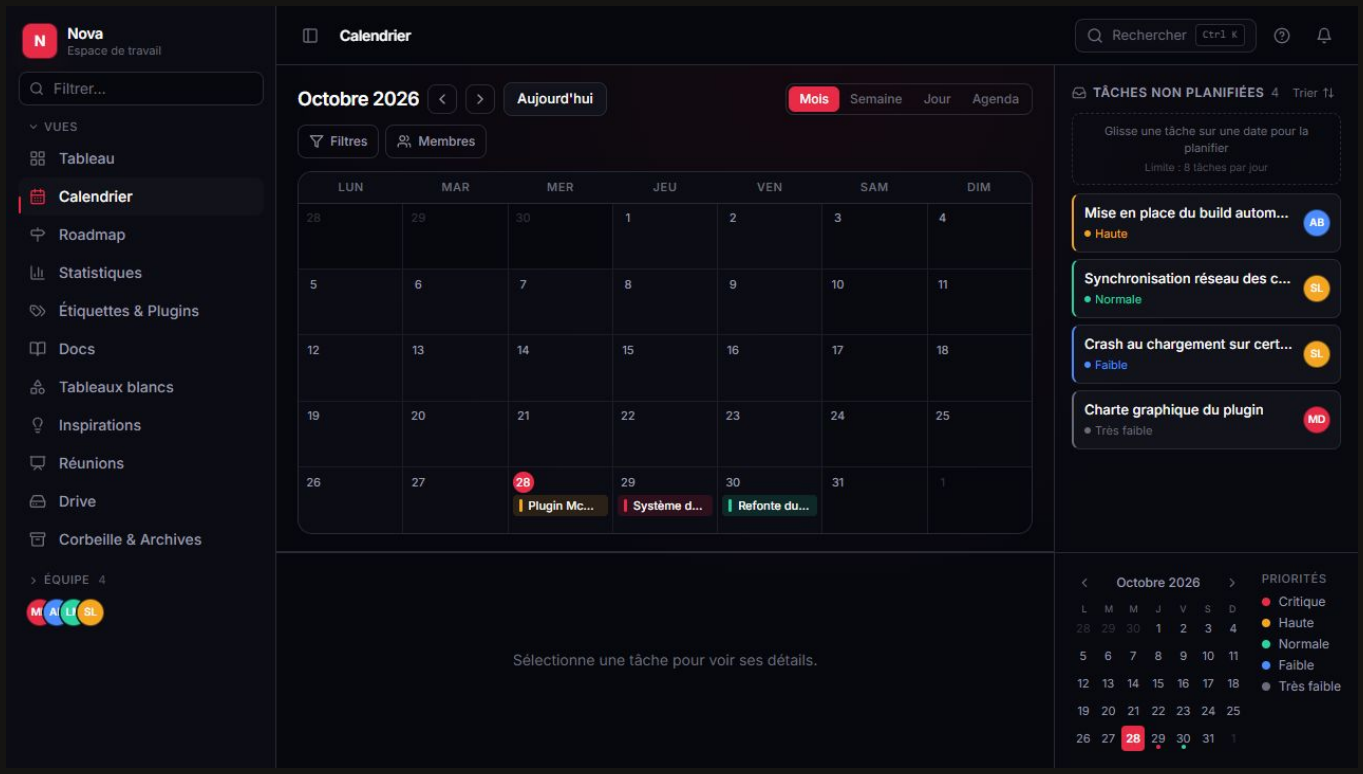
Tableau Kanban — mode local, exemples inclus dans les sources. L'interface porte encore le nom Nova.



Consultation d'une tâche — données d'exemple locales, contrôles d'édition désactivés.

04 / NAVIGATION

Planifier dans le calendrier



Calendrier — navigation observée en mode local, sans modification des tâches.

Ce qui a été observé

La navigation vers le calendrier, les tâches planifiées et le panneau des tâches non planifiées. L'édition, le glisser-déposer et les écritures cloud n'ont pas été exercés.

Application publique

La version publique affiche une connexion par pseudo et code. Aucune inscription ni saisie d'identifiants n'a été effectuée pendant cette inspection.

Essayer NEXARIS : kanban-e378.onrender.com

05 / CONTRIBUTION ET RECUL

Mon rôle, les limites, la suite

Contribution personnelle

Développement des interfaces, des fonctionnalités et de la logique de gestion de projets. Développement majoritaire du code, avec une assistance IA occasionnelle. Les responsabilités décrites proviennent des précisions de Romain Beka ; l'analyse des fichiers ne mesure pas l'auteur de chaque ligne.

Défis et points à vérifier

Les commentaires de Board.tsx documentent un problème de boucle de rendu pendant le glissement. La solution présente dans le code diffère les mutations jusqu'au dépôt et privilégie la détection sous le pointeur pour les colonnes vides. Le mécanisme de fusion traite aussi les états divergents. Aucun résultat de performance ou de test multiutilisateur n'est revendiqué ici.

Perspectives

Compléter la démonstration par un compte dédié et vérifier les parcours d'édition, la persistance après rechargement et les conflits entre utilisateurs dans un environnement de test autorisé.

Références

CV et précisions de Romain Beka : objectif, statut et contribution.

Sources NEXARIS : package.json, App.tsx, Board.tsx, router.ts, useBoardStore.ts, useCloudSync.ts, merge.ts et schéma SQL.

Inspection du 28 octobre 2026 : site public limité à la connexion ; captures du mode local isolé.