

Gestion de Freelancers

Objectif principal : Permettre à un freelance de gérer ses clients, projets, tâches, temps passé et facturation dans une seule application simple et efficace.

Bénéfices utilisateur :

- Suivre les tâches par client / projet.
- Mesurer précisément le temps travaillé (par tâche, par jour, par client).
- Générer des factures basiques à partir du temps ou des forfaits.
- Garder des notes, fichiers, liens importants par client / projet.
- Avoir une vue globale de la journée / semaine (tableau de bord).

Architecture Technique

- Frontend web : React JS
- Backend : Node.js + Express
- Base de données : MySQL
- Auth : JWT access + refresh
- Tests backend : Jest + Supertest
- Tests frontend : Jest + React Testing Library(Cypress / Playwright pour E2E)

Cibles & Scénarios d'Usage

2.1. Personas

Freelance développeur / designer / rédacteur

- Travaille avec 3–10 clients actifs.
- A plusieurs projets en parallèle.
- Doit justifier ses heures et envoyer des factures.

Petits collectifs / micro-agences (Bonus)

- Plusieurs freelances collaborent sur un même projet.
- Besoin de voir les temps par membre.

2.2. Scénarios Clés

- Créer un nouveau client → créer un projet → créer des tâches → suivre le temps → facturer.
- Voir la liste des tâches du jour / semaine avec statut.
- Filtrer les tâches par client / projet / statut.
- Générer une facture pour un client sur une période donnée.
- Exporter ses temps de travail en CSV. (bonus simple à implémenter)

Modules Fonctionnels

3.1. Module Authentification & Comptes

Création de compte freelance :

- Email, mot de passe (min. 8 caractères).
- Nom / prénom (ou pseudo).
- Devise par défaut (EUR, MAD, USD...).
- Connexion / déconnexion.
- Réinitialisation de mot de passe (par email ou flux simplifié).

1

Gestion du profil :

- Informations de facturation : nom légal, adresse, SIRET / ICE, TVA, logo (optionnel).

Gestion des tokens :

- JWT Access token (durée courte : 15–30 min).
- Refresh token (durée plus longue : 7–30 jours).
- Stockage sécurisé côté front (voir section 7).

3.2. Module Clients

CRUD client :

- Créer / lire / mettre à jour / supprimer (soft delete).

Champs client :

- Nom du client.
- Type (entreprise / particulier).
- Contact principal (nom, email, téléphone).
- Adresse de facturation.
- Notes internes.

Fonctionnalités :

- Liste des clients avec recherche + filtre (nom, type).
- Voir la fiche client :
 - Projets liés.
 - Factures liées.
 - Temps total facturé / non facturé (bonus).

2

Module Projets

Un projet appartient à un client et à un freelance (user).

Champs projet :

- Nom du projet.
- Client associé.
- Type de facturation :
 - TJM / taux horaire (ex: 50€/h).
 - Forfait (montant fixe de projet).
- Statut : actif / en pause / terminé.
- Date de début / date estimée de fin.
- Description.

Fonctions :

- CRUD projet.
- Lister les projets par client, par statut.
- Vue projet :
 - Liste des tâches.
 - Temps total passé.
 - Montant estimé / facturé (si lien avec factures).

Modules de Gestion

3.4. Module Tâches

Les tâches appartiennent à un projet.

Champs tâche : Titre, Description, Statut, Priorité, Date d'échéance, Temps estimé, Tags.

Fonctions : CRUD tâche, Changer le statut rapidement, Vue liste & vue Kanban (bonus), Filtrer par Client, Projet, Statut, Priorité, Date d'échéance.

3.5. Module Time Tracking (suivi du temps)

MVP : Saisie manuelle ou Timer (Démarrer / Arrêter, lier à une tâche).

Champs time entry : Tâche, Projet, Date, Heure de début / fin, Durée en minutes, Description courte, Facturé / non facturé.

Fonctionnalités : Liste des entrées de temps, Totaux (par jour, client, projet), Export CSV des temps (bonus).

3.6. Module Facturation basique

Objectif : générer des factures simples.

Entité facture : Numéro, Client, Date d'émission, Date d'échéance, Statut, Montant HT / TVA / TTC, Devise.

Lignes de facture : Description, Quantité, Prix unitaire, Total ligne.

Génération : À partir des time entries ou manuelle.

Fonctions : Marquer comme payée, Total facturé, Export PDF / CSV (bonus).

3.7. Module Notes & Documents

Notes globales ou par client / projet.

Champs : Titre, Contenu (texte riche ou markdown), Tags (optionnel).

Attachements (bonus) : Upload de fichiers liés.

3.8. Tableau de bord

Vue synthétique : Tâches du jour / en retard, Heures travaillées cette semaine, Total facturé ce mois, Clients actifs / projets actifs.

Possibilité de filtrer par période.

3.9. Paramètres & Préférences

Devise par défaut, Format de date, Langue (FR / EN – bonus).

Préférences de notifications : Rappel pour échéances de tâches, Rappel pour factures impayées.

3.10. Bonus Avancés (V2+)

Collaboration : Plusieurs utilisateurs, Rôles.

Intégrations : Calendrier, Export comptable.

Autres : Mode offline, Mode "Pomodoro".

Exigences Non Fonctionnelles



Sécurité :

- Hash des mots de passe (bcrypt).
- JWT signé, rotation des refresh tokens.
- Validation des données (backend + frontend).



Performance :

- Endpoints optimisés avec pagination pour les listes (clients, projets, tâches).



Qualité :

- Couverture de tests minimale :
 - Backend : 60–70%.
 - Frontend : tests sur les écrans critiques.



UX / UI :

- Responsive.
- Dark / light mode (bonus).

Modèle de Données

User	id, email, password_hash, first_name, last_name, currency, company_name, address, tax_id, created_at, updated_at
Client	id, user_id, name, type, contact_name, contact_email, contact_phone, billing_address, notes, is_archived, created_at, updated_at
Project	id, user_id, client_id, name, description, billing_type, hourly_rate, fixed_amount, status, start_date, end_date_estimated, created_at, updated_at
Task	id, project_id, title, description, status, priority, due_date, estimated_hours, created_at, updated_at
TimeEntry	id, user_id, project_id, task_id, date, start_time, end_time, duration_minutes, description, is_billed, invoice_id, created_at, updated_at
Invoice	id, user_id, client_id, number, issue_date, due_date, status, currency, total_ht, total_tva, total_ttc, created_at, updated_at
InvoiceItem	id, invoice_id, description, quantity, unit_price, total, project_id (Optionnel)
Note	id, user_id, client_id, project_id, title, content, created_at, updated_at
RefreshToken	id, user_id, token, expires_at, created_at, revoked_at

Bonus V2 : Tag, TaskTag, Attachment, Notification.

Gestion des Tokens & API Endpoints

7.1. Auth Flow

- **Login** : POST /auth/login avec email + password. Vérification du mot de passe. Réponse : accessToken (JWT, 15–30 min), refreshToken (JWT ou token opaque, 7–30 jours).
- **Refresh** : POST /auth/refresh avec refreshToken. Vérification (non expiré, non révoqué). Renvoi d'un nouveau accessToken.
- **Logout** : POST /auth/logout → invalide / supprime le refresh token côté serveur.

7.2. Stockage Côté Front

- **Access token** : En localStorage.
- **Refresh token** : En localStorage (Idéalement en cookie HTTPOnly)

7.3. Middleware Backend

Middleware auth qui : Lit le header Authorization: Bearer . Vérifie la signature du JWT. Charge le user_id dans req.user. Refuse si invalide.

8. API Endpoints (REST)

- **8.1. Auth** : /register, /login, /refresh, /logout, /me
- **8.2. Clients** : GET, GET/:id, POST, PUT/:id, DELETE/:id
- **8.3. Projects** : GET, GET/:id, POST, PUT/:id, DELETE/:id
- **8.4. Tasks** : GET, GET/:id, POST, PUT/:id, PATCH/:id/status, DELETE/:id
- **8.5. Time Entries** : GET, POST, PUT/:id, DELETE/:id, POST/start, POST/:id/stop
- **8.6. Invoices** : GET, GET/:id, POST, PUT/:id, POST/:id/mark-paid, GET/:id/download
- **8.7. Notes** : GET, GET/:id, POST, PUT/:id, DELETE/:id
- **8.8. Dashboard** : GET /summary

Structure de Pages et Tests

9.1. Pages Principales

- /login, /register
- /dashboard
- /clients, /clients/:id
- /projects, /projects/:id
- /tasks (vue globale + filtres)
- /time-tracking (vue de la journée / semaine + saisie / timer)
- /invoices, /invoices/:id
- /notes
- /settings

9.2. Composants Clés

- Layout général (sidebar + header).
- Composant ClientList, ClientForm.
- ProjectList, ProjectForm.
- TaskList, TaskForm, TaskKanbanBoard (bonus).
- TimeEntryList, TimeEntryForm, TimerWidget.
- InvoiceTable, InvoiceForm, InvoicePreview.
- DashboardCards (statistiques).
- FiltersBar réutilisable (recherche, filtres, tri).

10.1. Backend

Unit tests :

- Services de : création / update de client, projet, tâche, calcul des montants de facture, génération de numéro de facture.
- Utils : formatage des dates / montants, calcul du temps total.

Tests d'intégration (API) :

- Auth : register / login / refresh / accès à /me.
- Clients : création, update, suppression, pagination.
- Tâches : création, changement de statut.
- Time entries : création, filtrage par période.
- Factures : création, changement de statut en paid.

Outils : Jest + Supertest, base test en mémoire (SQLite)

10.2. Frontend

Unit / integration tests (React Testing Library) :

- Composant Login : saisie email / password, appel API mocké.
- Dashboard : affichage des stats venant d'un mock d'API.
- TaskList : filtre par statut, changement d'état visuel.
- TimeTracking page : ajout d'une entrée de temps, affichage dans la liste.

E2E (bonus) :

- Scénario complet : Login, Création client + projet + tâche, Ajout de temps, Génération d'une facture, Vérification sur la page factures.

Outils : Cypress / Playwright.