

TwitchPlanner - Mini rapport (B3)

Application de planification de streams Twitch

Auteur : Shamiul Alam Upom | Date : 13/02/2026 | Repo GitHub : [lien public](#)

Technologies: Node.js/Express, React, TailwindCSS, PostgreSQL, Prisma, JWT

1. Contexte et objectifs

TwitchPlanner permet aux streamers de créer un planning hebdomadaire de stream, d'ajouter des événements (jeu, horaire, titre) et d'exporter le planning sous forme d'image (PNG) pour le partager.

Fonctionnalités livrées :

- Inscription / connexion (JWT) et pages protégées.
- Profil utilisateur (informations de base).
- CRUD de planifications et CRUD d'événements (plusieurs événements par jour).
- Export PNG du planning.

2. Architecture et choix techniques

- Front-end : React (pages, composants, modales, rendu du planning).
- Back-end : Express (API REST, règles métier, gestion d'erreurs).
- ORM : Prisma (migrations, requêtes typées).
- BDD: PostgreSQL (relations + constraints).

Flux : React (UI) -> API Express -> Prisma -> PostgreSQL

3. MCD (Modèle Conceptuel)

Entités :

Entité	Attributs (principaux)
Utilisateur (User)	id, email, passwordHash, twitchUrl, logoUrl
Planning	id, userId, name, weekStart, weekEnd, (optionnel: couleurs)
Evènement (Event)	id, planningId, gameName, title?, gameImageUrl?, startsAt, endsAt

Cardinalités :

- User 1 -> N Plannings
- Planning 1 -> N Events

Schéma :

USER (1) ---- (N) PLANNING (1) ---- (N) EVENT

4. MPD (Modèle Physique) - PostgreSQL

Tables (résumé) :

4.1 users

Colonne	Type	Contraintes
id	UUID	PK
email	TEXT	UNIQUE, NOT NULL
password_hash	TEXT	NOT NULL

4.2 plannings

Colonne	Type	Contraintes
id	UUID	PK
user_id	UUID	FK -> users.id, NOT NULL
week_start	DATE	NOT NULL
week_end	DATE	NOT NULL

4.3 events

Colonne	Type	Contraintes
id	UUID	PK
planning_id	UUID	FK -> plannings.id, NOT NULL
game_name	TEXT	NOT NULL
starts_at	TIMESTAMP	NOT NULL
ends_at	TIMESTAMP	NOT NULL

Règle : ends_at > starts_at (validation côté API).

5. Difficultés et solutions

- **Problèmes Prisma / TypeScript** : au début, Prisma n'était pas bien généré. On a eu une erreur avec PrismaClient et on a dû vérifier les versions (prisma et @prisma/client) et relancer prisma generate.
- **Export PNG** : au début, l'export ne téléchargeait pas. Ensuite, html2canvas a cassé à cause de la couleur oklch. On a remplacé par html-to-image, et l'export marche.
- **Export PNG** : compatibilité CSS/images, tests et réglages (crossOrigin).

6. Compétences mises en pratique

- Conception relationnelle (MCD/MPD) et contraintes métier.
- API REST (validation, gestion d'erreurs, statuts HTTP).
- Authentication JWT et protection des routes.
- React (state, composants, modales) et intégration Tailwind.
- Génération d'un export PNG à partir d'un rendu HTML. (html-to-image et html2canvas)