**🚀 Quickstart — Realtime Quiz**

**1. Démarrer PostgreSQL**

Sous Windows (service installé nativement) :

* Vérifie que le service Postgres tourne :  
  Win + R → services.msc → **PostgreSQL 16** → doit être en statut *Running*.
* Sinon, démarre-le à la main.

Tester la connexion :

& 'C:\Program Files\PostgreSQL\16\bin\psql.exe' -h localhost -U dev -d quiz -W -c "\conninfo"

(Mot de passe : dev)

**2. Backend (Fastify + Socket.IO)**

cd server

# Installer les dépendances (si nouvel environnement)

npm install

# Vérifier Prisma

npx prisma validate

# Appliquer les migrations

npx prisma migrate dev

# (Ré)initialiser la DB et lancer le seed

npx prisma db seed

# Lancer le backend

npm run dev

➡ Le backend écoute sur http://localhost:3000  
Test rapide :

Invoke-RestMethod http://localhost:3000/health

# doit renvoyer {"ok":true}

**3. Frontend (React + Vite)**

cd web

# Installer les dépendances (si nouvel environnement)

npm install

# Lancer le frontend

npm run dev

➡ Ouvre l’URL donnée par Vite (souvent http://localhost:5173).

**4. Jouer une partie de test**

1. Récupérer le **code de jeu** généré par le seed :
   * Via Prisma Studio :
   * npx prisma studio

→ ouvre <http://localhost:5555>, clique sur **Game**, copie le champ code.

* + Ou via Postgres :
  + SELECT code FROM "Game" ORDER BY "createdAt" DESC LIMIT 1;

1. Ouvrir **2 onglets** du frontend → saisir le même code + noms différents → Join.
2. Dans un des onglets → cliquer sur **Start (host)**.
3. Observer :
   * Timer qui défile.
   * Bonne réponse affichée à la fin du round.
   * Scoreboard mis à jour.
   * Auto-avance vers la question suivante.
   * Game over avec classement final.

**5. Résumé des commandes essentielles**

* **Backend dev server**
* cd server
* npm run dev
* **Frontend dev server**
* cd web
* npm run dev
* **Reseed la base (nouveau code de jeu)**
* cd server
* npx prisma migrate reset
* **Explorer/modifier la DB**
* cd server

npx prisma studio