**Fejlesztői dokumentáció – CheckPoint**

### **Projekt neve**: CheckPoint **Készítők neve**: Csillag Csaba, Nagy Dorina Laura

### Cél

Funkcionális, reszponzív webalkalmazás, amely az alábbi core funkciókat valósítja meg:

* Felhasználói regisztráció és hitelesítés (JWT)
* Bejelentkezés / kijelentkezés
* Főoldal / tartalom letöltése
* Kapcsolatfelvétel űrlap  
  A rendszer célja egy beléptetési rendszer prototípusának megvalósítása modern technológiákkal.

### Fejlesztői környezet

* **IDE:** Visual Studio Code
* **Verziókövetés:** Git + GitHub
* **API-tesztelés:** Flashpost
* **Konfiguráció:** .env alapú környezetkezelés (DATABASE\_URL, SECRET\_KEY stb.)

### Frontend Technológiai stack

* React (w/ Vite)
* TypeScript
* React Router
* Tailwind CSS
* Lucide Icons
* Fetch API (async kommunikáció)

### Backend Technológiai stack

* NestJS (TypeScript alapú REST API)
* MySQL (relációs adatbázis)
* Prisma ORM
* JWT (authentikáció)
* Argon2 (jelszóhash-elés)
* Swagger – automatikus API dokumentáció (/api endpointon)

**Telepítés/Futtatás**

Frontend  
*git clone* [*https://github.com/Dorolesz/CheckPoint\_Frontend.git*](https://github.com/Dorolesz/CheckPoint_Frontend.git)

*cd CheckPoint\_Frontend*

*npm install*  (Függőségek telepítése)

*npm run dev*

Backend

*cd CheckPoint\_Backend  
npm install (Függőségek telepítése)  
# .env fájl létrehozása****DATABASE\_URL="mysql://root@localhost:3306/CheckPoint"***

***SECRET\_KEY="yourSuperSecretKey"***

***# Adatbázis migráció és feltöltés:***

*npx prisma migrate dev --name init*

*npx prisma db seed  
# vagy alternatíva:*

*npx ts-node prisma/seed.ts*

### *npm run start:dev* Auth és biztonság

* JWT token alapú authentikáció
* Hash-elés: Argon2
* Role alapú jogosultságkezelés későbbi bővítésre előkészítve

**Swagger API dokumentáció**

A localhost:3000/api címen érhető el a teljes backend endpoint dokumentáció NestJS Swagger integrációval.