

Biztonságos Webapp Fejlesztés Ellenőrzőlista

Mind a 10 lépés egyetlen áttekintő lapon — Minőségbiztosítás kezdőknek és haladóknak

☐ 01

ALAPOK

Kezdj egyszerűen (Adatbázis nélkül)

Kezdőként építs adatbázis- és user-mentes kliensoldali appot (pl. kalkulátor, konverter). Amit nem tárolsz a szerveren, azt nem lehet ellopni.

☐ 06

BAAS

Használd bevált Managed BaaS-t

Ne kódolj saját auth-ot! Használd Supabase/Firebase-t (Auth & RLS), Stripe-ot (Fizetés), Resend-et (Email) és Vercel-t (Hosztging).

☐ 02

BIZTONSÁG

Titkok és környezeti változók (.env)

Soha ne égesd be az API kulcsokat a kódba! A .gitignore-ba az első perctől vedd fel a .env-et, és használd .env.example mintafájlt.

☐ 07

VERZIÓ

Git mint mentési pont (Kis lépések)

Amint egy funkció működik, azonnal commitolj! Ha az AI elrontja a kódot, másodpercek alatt visszaállhatsz a működő állapotra.

☐ 03

ARCHITEKTÚRA

3-Layer Architektúra (AGENTS.md)

Rögzíts szabályokat a gyökérben: válaszd szét a direktívákat (directives/), a döntéshozatalt és a szkripteket (execution/).

☐ 08

AUDIT

Cross-Model Code Review

Ha az egyik modellel (pl. Gemini) elkészült a kód, add át egy másiknak (pl. Claude) biztonsági audit prompttal a vakfoltok kiszűrésére.

☐ 04

ESZKÖZ

Kontextus-motor és optimalizálás

Használd vexp.dev-et: az AST függőségi gráfja csak a releváns kódot adja át az AI-nak (akár -85% token), és mechanikusan ellenőrzi.

☐ 09

BIZTONSÁG

Automata Biztonsági Pásztázás

Futtass SAST vizsgálatot a GitHub repón a Rafter.so-val: megtalálja a sérülékenységeket és azonnal alkalmazható javítókédot ad.

☐ 05

PROMPTING

Mindig kérj Tervet kódírás előtt

Bármely pillanatban állítsd meg az AI-t és kérj 4-5 lépéses megvalósítási tervet az architektúráról, mielőtt kódot írna.

☐ 10

KONTROLL

Költségkontroll & Red Flag figyelés

Állíts be spend limiteket és rate limitet. Állítsd le az AI-t 1000+ soros fájloknál, loop hibáknál és eltűnő funkcióknál.

VIBE CODING RED FLAGS:

- 1000+ soros monolit fájlok (nincs moduláris felépítés)
- AI végtelen hibajavítási hurokban forog (Stop & új kontextus)
- Korábbi funkciók eltűnnek / Felesleges overengineering
- Hardcoded jelszavak és API kulcsok a kódban

GYORS PROMPT ARANYSZABÁLYOK:

- Írj 4 lépéses tervet kódírás előtt és várj jóváhagyásra!
- Egyszerre csak egy funkciót kérj és tesztelj!
- Auditáld a kódot OWASP és jogosultsági szempontból!
- Minden sikeres teszt után azonnal: Git Commit!