

Architecture Decision Records

1.

Titlu: Tehnologii utilizate: ASP.NET Core si React

Status: acceptat

Context: ASP.NET ofera suport built-in pentru REST apis si Swagger pentru verificare rapida a endpoint-urilor, integrare simpla cu baza de date

Microsoft SQL Server, iar React este o librerie populara ce faciliteaza implementarea unui UI responsive

Decizie: Vom construi aplicatia folosind ASP.NET Core pentru backend si React pentru frontend intrucat toti membri echipei sunt familiari cu aceste tehnologii

Consecinte: Vom avea aplicatii separate pentru frontend si backend

Respectare: Am colaborat pe Github pentru a asigura consistenta codului

2.

Titlu: RESTful API Design

Status: acceptat

Context: Avem nevoie de o metoda clara si consistenta de comunicare intre frontend si backend

Decizie: Vom folosi principii RESTful pentru a construi endpoint-urile intrucat este un standard acceptat, clar, cunoscut de toti membri echipei

Consecinte: Usor de inteles si de utilizat de catre programatori

Respectare: Code review

3.

Titlu: Autentificare cu Identity

Status: acceptat

Context: Aplicatia are nevoie de autentificare sigura pentru a proteja datele utilizatorilor

Decizie: Vom folosi framework-ul ASP.NET Core Identity cu jwt intrucat are suport pentru hashing al parolelor, role management si impune parole sigure

Consecinte: Simplifica implementarea autentificarii, membri echipei trebuie sa fie familiari cu framework-ul

Respectare: Clasa pentru utilizatori extinde clasa IdentityUser

4.

Titlu: Utilizat Entity Framework Core

Status: acceptat

Context: Backend-ul trebuie sa poata comunica eficient si usor cu baza de date

Decizie: Vom folosi ORM-ul Entity Framework Core pentru interactiunile cu baza de date intrucat simplifica accesul utilizand linq in loc de sql

Consecinte: Minimizeaza riscul de sql injection

Respectare: Code review