Marcus Lehm

Projektportfolio

{ MarcusLehm.se }

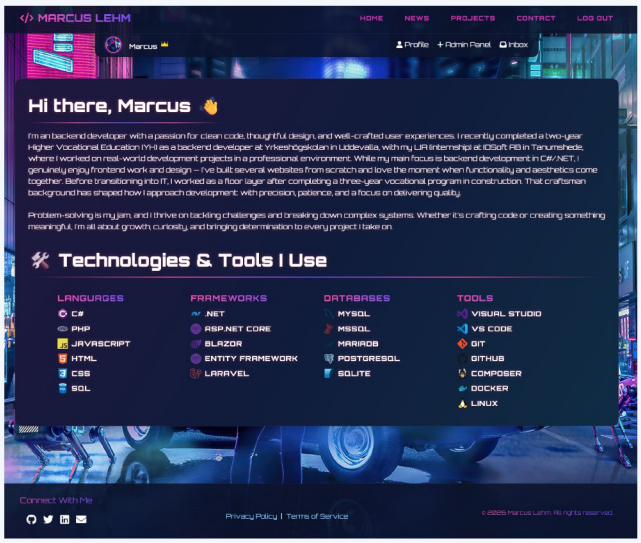
01 WEBB & INNEHÅLLSHANTERING

Personlig webbplats med administration och användarflöden

PHP · Laravel · Blade · MariaDB · JavaScript · PHPUnit · Docker

En personlig webbplats som samlar projekt, nyheter och kontakt på ett ställe. Utöver den publika presentationen finns användarkonton, profiler, kommentarer och privata meddelanden. Syftet är att kunna presentera mitt arbete och samtidigt hantera innehåll och kommunikation i en egen webbapplikation, med olika behörigheter för besökare, användare och administratörer.

preview



Startsidan med profil, navigation och tekniköversikt.

// Vad jag gjorde

Jag utvecklade webbplatsens gränssnitt och backend för publicering, konton och kommunikation. Arbetet omfattade Eloquent-modeller, formulärvalidering, åtkomstregler och Google-inloggning. Jag ansvarade även för implementationen av privata meddelanden med bilagor och köade e-postnotiser med användarvalda inställningar. Tester för bland annat behörigheter och köhantering ingick i arbetet med att upptäcka fel och kontrollera funktionaliteten.

// Resultat

En sammanhållen webbapplikation för presentation, publicering och användarkommunikation. Projektet visar fullstackarbete med relationsdata, åtkomstkontroll och bakgrundsjobb samt tester av centrala användarflöden.

// Tekniska höjdpunkter

01 Behörigheter

Policies och middleware styr vem som får ändra projekt, innehåll och användaruppgifter.

02 Meddelanden

Transaktioner och klienttoken hanterar meddelanden och bilagor samt motverkar dubbla utskick vid upprepade anrop.

03 E-post i kö

Notiser köas efter databasens commit, med återförsök och kontroll av mottagarens aktuella inställningar.

Marcus Lehm

Projektportfolio

{ Office Event Hub }

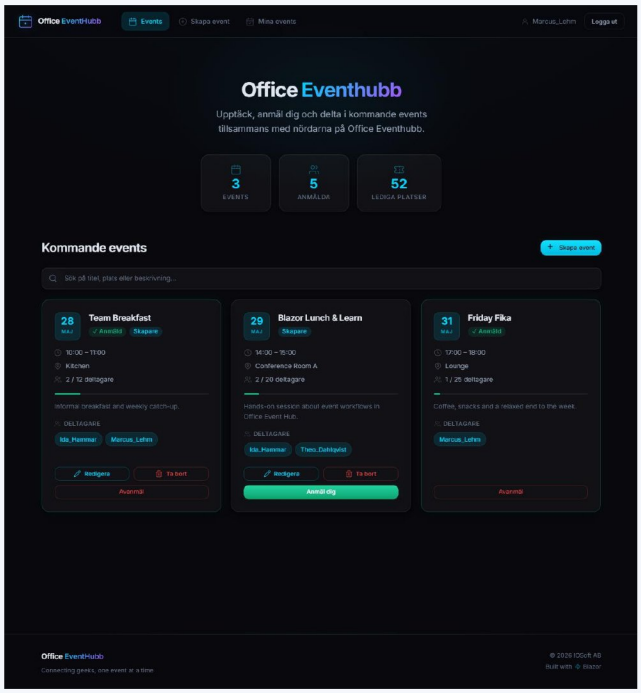
02 C# / .NET · EVENTHANTERING

Aktiviteter, deltagare och anmälningar i samma webbapp

C# · .NET 10 · Blazor · EF Core · SQLite · ASP.NET Identity · xUnit

En webbapp för att samla aktiviteter och anmälningar på en arbetsplats. Användaren kan hitta kommande event, se deltagare och lediga platser samt anmäla eller avanmäla sig. Arrangören kan skapa och ändra aktiviteter. Syftet är att ge en gemensam överblick över vad som händer och vem som deltar.

preview



Eventöversikt med deltagare, lediga platser och anmälan.

// Vad jag gjorde

Jag byggde eventflödet från datamodell och affärsregler till Blazor-gränssnitt och HTTP-endpoints. Jag ansvarade för implementationen av konton, anmälningar och arrangörens hantering av deltagare. Arbetet omfattade separata lager för domän, applikation och infrastruktur, EF Core och automatiska uppdateringar i gränssnittet. Jag arbetade också med tester av regler, användningsfall och HTTP-flöden för att kontrollera funktionaliteten.

// Resultat

En webbapp där aktiviteter kan skapas, sökas och administreras med deltagarlistor och anmälningar. Projektet visar hur affärsregler, datalagring och ett interaktivt .NET-gränssnitt kan kopplas samman och testas.

// Tekniska höjdpunkter

01 Tydliga lager

Domänregler och användningsfall är separerade från databas, webbgränssnitt och API.

02 Regler för deltagande

Domänen hanterar maxantal och dubbelanmälningar. Ett unikt databasindex skyddar kombinationen event och användare.

03 Tester på flera nivåer

Domän- och applikationstester kompletteras av HTTP-tester med WebApplicationFactory och SQLite.

Marcus Lehm

Projektportfolio

{ PlaySphere }

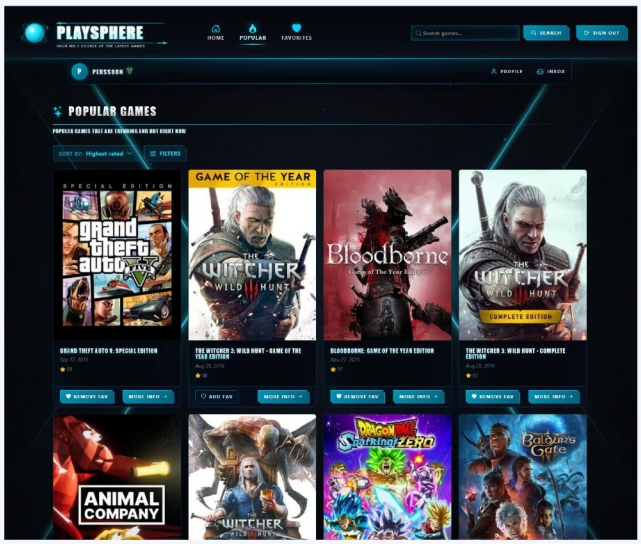
03 C# / .NET · API-INTEGRATION

Spelkatalog med filtrering och personliga favoriter

C# · .NET 10 · Blazor · EF Core · SQL Server · IGDB API · xUnit

En spelkatalog som hämtar information från IGDB och låter användaren upptäcka, söka efter och läsa om spel. Katalogen kan filtreras efter bland annat genre, plattform, utgivningsår och betyg. Inloggade användare kan spara favoriter. Syftet är att samla extern spelinformation och personliga urval i ett och samma gränssnitt.

preview



Populära spel med sortering, filter och favoritknappar.

// Vad jag gjorde

Jag utvecklade Blazor-gränssnittet och backendlagren som kopplar spelkatalogen till IGDB. Jag ansvarade för implementationen av sökning, filtrering, sortering och sidindelning samt konton och databaslagrade favoriter. Arbetet omfattade tokenhantering, cache och tillståndshantering i gränssnittet. Jag använde tester för att kontrollera API-klient, tokenförnyelse, filter och favoriter, med fokus på hur delarna fungerar tillsammans.

// Resultat

En sökbar spelkatalog med detaljsidor och sparade favoriter per användare. Projektet visar arbete med externa API:er, autentisering, relationsdata och tillstånd i ett interaktivt webbgränssnitt.

// Tekniska höjdpunkter

01 Extern speldata

En separat HTTP-klient översätter IGDB-data till egna modeller. Åtkomsttoken återanvänds och förnyas vid behov.

02 Cache & filtrering

Cachelagrade katalogdata och antalsträffar kombineras med filter, sortering och sidindelning som styrs via URL:en.

03 Personliga favoriter

Identity-konton kopplas till favoriter i SQL Server, med unikt index som förhindrar dubbletter.

Marcus Lehm

Projektportfolio

{ ÄsperödBot }

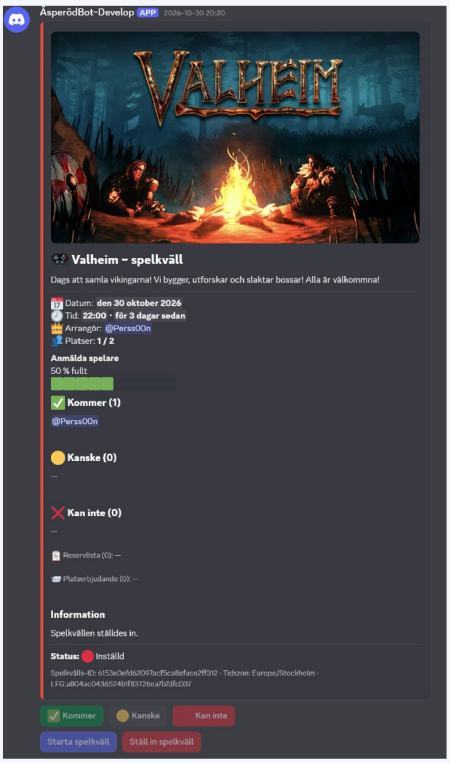
04 C# / .NET · DISCORD & AUTOMATION

Spelkvällar och spelgrupper direkt i Discord

C# · .NET 10 · Discord.Net · EF Core · PostgreSQL · xUnit · Testcontainers

En Discord-bot för att planera spelkvällar och samla spelare i grupper. Medlemmar kan anmäla sig, stå på reservlista och få påminnelser utan att lämna Discord. När en spelkväll startar förs deltagarna över till grupphanteringen. Syftet är att samla planering, deltagarstatus och privata röstkanaler i serverns befintliga miljö.

preview



Spelkväll med deltagarstatus, reservlista och åtgärdsknappar.

// Vad jag gjorde

Jag utvecklade kommandon och interaktiva flöden för spelgrupper, spelkvällar och en inbyggd hjälpguide. Jag ansvarade för implementationen av beständig deltagarhantering, reservlistor, påminnelser och övergången från spelkväll till spelgrupp. Arbetet omfattade synkronisering mellan databasen och Discord samt samtidiga åtgärder. Jag arbetade också med integrationstester för att kontrollera beteendet vid omstarter, databasavbrott och upprepade startförsök.

// Resultat

Ett sammanhängande flöde från anmälan till spelgrupp och röstkanal. Projektet visar asynkron
backendlogik, beständigt tillstånd och hantering av fel vid integration med en extern tjänst.

// Tekniska höjdpunkter

01 Transaktioner & samtidighet

Start av spelkväll och ny grupp sparas i samma transaktion. PostgreSQL-lås samordnar samtidiga ändringar.

02 Återhämtning & tester

Sparade operationer synkroniserar Discord efter avbrott. Testcontainers-tester kontrollerar databasavbrott och att upprepad start inte skapar flera grupper.