**Projektdokumentation: C2C Markedsplads for Bærbare PC'er**

**Projektnavn:** ***LapSwap***   
**Team:** Bjarke, Andreas & Frederik

**1. Indledning og Overordnet Arkitektur**

**1.1 Projektets Formål og Vision**

Projektets formål var at udvikle "LaptopMarket", en webbaseret C2C (Consumer-to-Consumer) markedsplads, der specialiserer sig i køb, salg og bytte af nye, brugte og renoverede bærbare computere. Visionen var at skabe en intuitiv og sikker platform, hvor private brugere nemt kunne oprette en konto, administrere deres egne annoncer (oprette, redigere, slette) og finde den perfekte computer via et detaljeret og brugervenligt filtreringssystem.

**1.2 Teknisk Arkitektur og Implementering**

Den endelige applikation, "LapSwap", blev udviklet på **.NET 8-platformen** ved hjælp af **Blazor Server** som hostingmodel. Dette arkitektoniske valg sikrer, at al C#-kode og forretningslogik eksekveres på serveren, mens UI-opdateringer håndteres effektivt via en vedvarende SignalR-forbindelse.

* **Database:** Som planlagt blev en PostgreSQL-database hos **Neon.Tech** anvendt.
* **Dataadgang:** I tråd med kravet om **IKKE at bruge en ORM**, blev al databaseinteraktion implementeret direkte med **Npgsql-driveren**, indkapslet i et Repository Pattern for at sikre en ren og vedligeholdelsesvenlig kodebase.
* **Frontend:** Designet blev bygget med **Bootstrap 5.1**, hvilket sikrede et responsivt og moderne layout, der stemte overens med de oprindelige designskitser.

**2. Databasedesign og Dataadgangslag**

**2.1 Fra Vision til Realitet: Datamodellen**

Den oprindelige vision skitserede to primære tabeller, users og product\_models. Under udviklingen blev dette design raffineret for bedre at understøtte platformens funktionalitet, især behovet for flere billeder pr. annonce. Den endelige datamodel består af tre kerneentiteter:

* **User:** Indeholder brugeroplysninger (navn, email, kontaktinfo osv.).
* **Listing:** Indeholder alle data for en annonce (titel, mærke, model, pris, specifikationer) med en UserId fremmednøglerelation til User.
* **ListingImage:** Indeholder stien til et billede med en ListingId fremmednøglerelation til Listing.

Dette design følger normaliseringsprincipper og skaber klare én-til-mange-relationer: En User kan have mange Listings, og et Listing kan have mange ListingImages.

**2.2 Repository Pattern: Struktureret Dataadgang uden ORM**

For at abstrahere og centralisere databasekald blev **Repository Pattern** implementeret. Dette adskilte UI-komponenterne (Razor-sider) fra den rå SQL-logik.

**Nøgleimplementeringer i ListingRepository:**

* **Effektiv Datahentning (LEFT JOIN):** For at undgå det klassiske "N+1 selects"-problem, blev en metode implementeret, der henter annoncer og deres tilhørende billeder i **én enkelt databaseforespørgsel** ved hjælp af en LEFT JOIN. Dette forbedrede performance markant ved visning af annoncelister.
* **Atomiske Operationer (Transaktioner):** Sletning af en annonce kræver sletning af både billedreferencer og selve annoncen. Disse operationer blev indkapslet i en NpgsqlTransaction. Dette garanterer **dataintegritet**: enten lykkes begge operationer, eller også rulles de begge tilbage ved fejl, hvilket forhindrer forældreløse data i databasen.

**3. Brugergrænseflade (UI) og Sidens Funktionalitet**

Den oprindelige vision for hver side blev i høj grad realiseret. Her er en gennemgang af, hvordan designidéerne blev omsat til funktionelle komponenter.

**3.1 Global State og Navigation (UserStateService)**

En central udfordring var at dele login-status på tværs af alle komponenter. Dette blev løst elegant med en Scoped service, UserStateService. Når en bruger logger ind, opdaterer servicen sin tilstand og udløser et OnChange-event. MainLayout.razor (som indeholder den øverste navigationsbar og footeren) abonnerer på dette event og gen-renderer sig selv, så "Log ind"-knappen øjeblikkeligt skifter til en profilikon.

**3.2 Sider og deres Sammenhæng**

* **Forside (HomePage.razor) og Annonceoversigt (ListingsPage.razor)**
  + **Vision:** En indbydende forside med et centralt søgefelt og "Premium Annoncer". En separat side til at vise alle annoncer med avancerede filtre i venstre side og et grid af resultater til højre.
  + **Implementering:** Visionen blev fulgt tæt. Søgefeltet på forsiden navigerer til ListingsPage.razor med en query-string. Filtreringsmenuen på ListingsPage giver brugeren fuld kontrol over de viste resultater, og knapperne "Anvend Filtre" og "Nulstil" giver en intuitiv brugeroplevelse.
* **Opret/Rediger Annonce (CreateListing.razor)**
  + **Vision:** En dedikeret formular til at oprette en ny annonce.
  + **Implementering:** En enkelt, genanvendelig komponent, CreateListing.razor, blev skabt til både at **oprette og redigere** annoncer. Komponentens tilstand styres af en URL-parameter (?editId=...). Hvis editId er til stede, hentes den eksisterende annonce, og formularen forud-udfyldes. Dette er et stærkt eksempel på genbrug af kode.
* **Brugerkonto (UserProfile.razor)**
  + **Vision:** Et dashboard med en venstremenu til navigation mellem personlige oplysninger, aktive annoncer, osv.
  + **Implementering:** Siden blev bygget som et centralt dashboard. En lokal variabel og en switch-sætning bruges til at vise forskellige sektioner (f.eks. "Personlige Oplysninger", "Mine Annoncer") uden at skulle navigere til helt nye sider, hvilket skaber en flydende "single-page app"-oplevelse. Brugerens aktive annoncer vises i en tabel med Rediger- og Slet-ikoner, der navigerer korrekt.
* **Login (LoginPage.razor) og Opret Bruger (RegisterPage.razor)**
  + **Vision:** Simple og intuitive formularer til autentificering.
  + **Implementering:** Begge sider blev implementeret som beskrevet. Navigationsflowet blev sikret, så en bruger, der forsøger at tilgå en beskyttet side (f.eks. "Opret Annonce"), bliver viderestillet til login-siden først og derefter sendt til den oprindeligt ønskede destination efter succesfuldt login.
* **Annonce-detaljeside (ListingDetailsPage.razor)**
  + **Vision:** En detaljeret visning af én annonce med billedgalleri, specifikationer, sælgerinfo og lignende annoncer.
  + **Implementering:** Siden blev implementeret som specificeret og fungerer som slutpunktet for de fleste brugerrejser. Den viser alle relevante oplysninger på en overskuelig måde og linker videre til andre relevante annoncer.

**4. Sikkerhedsanalyse og Læringspunkter**

**4.1 Adgangskodehåndtering**

Som en kritisk sikkerhedsforanstaltning følger applikationen best-practice ved at hashe alle adgangskoder med **BCrypt.Net**. Dette sikrer, at adgangskoder ikke gemmes i klartekst og er beskyttet mod rainbow table-angreb, selv i tilfælde af et databasetab.

**4.2 Identificeret Sårbarhed og Løsningsforslag**

Under analysen af den færdige applikation blev der identificeret en potentiel sikkerhedssårbarhed:

* **Problem:** En autentificeret bruger kunne teoretisk set manipulere URL'en (f.eks. /CreateListing?editId=123) og dermed få adgang til at redigere eller slette en annonce, der tilhørte en *anden* bruger. Den nuværende implementering validerede kun annoncens eksistens, ikke dens ejerskab.
* **Løsningsforslag:** Autorisering skal håndhæves på server-siden i dataadgangslaget.
  1. Metoderne UpdateAsync og DeleteAsync i IListingRepository skal udvides til også at kræve ID'et på den **aktuelt indloggede bruger**.
  2. Metoden bør derefter verificere, at operationen rent faktisk påvirkede en række. Hvis ikke, betyder det, at brugeren ikke havde tilladelse, og en undtagelse (f.eks. UnauthorizedAccessException) bør kastes.

Dette læringspunkt understreger vigtigheden af altid at validere ejerskab på serversiden og aldrig stole på klient-input alene.

**5. Konklusion**

Projektet lykkedes med at omsætte en detaljeret vision til en funktionel og teknisk solid webapplikation. Ved at anvende Blazor Server og en ren dataadgangsstrategi uden ORM blev målene om sikkerhed og kontrol over SQL opfyldt. Designet af brugergrænsefladen og den logiske sammenhæng mellem siderne blev realiseret som planlagt, hvilket resulterede i en intuitiv brugeroplevelse. Den afsluttende analyse afslørede en robust platform, men også et kritisk læringspunkt inden for sikkerhed, hvilket demonstrerer en moden og reflekteret udviklingsproces.