

Arkitekten

Kravspecifikation

Produktkrav

1. Ska kunna registrera sig som användare
2. Måste vara inloggad för att se sina projekt
3. Varje projekt ska ha en sida, ska gå att lägga till produkter
4. Ska gå att lägga till nya projekt
5. Kunna se totalpriset för projektet

Lösning

1. ASP.NET Core Identity är implementerad så användaren skall kunna registrera och logga in
2. En check görs mha User.Identity.Name som jämförs med sparad username i varje projekt. På detta sätt visas bara projekt om rätt användare är inloggad (frameworken kanske har egen lösning som är bättre???)
3. Sida för varje projekt implementerad mha action/id
4. Projekt sidan har en input som läser av produkt. Produkt Id och Projekt Id skickas till Bind och lagras i databasen.
5. Totalpriset är implementerad för varje projekt och uppdateras när en produkt läggs till.

Upplägg

Hemsidan är upplagt enligt MVC, detta för att det blir lättare att koda samt felsöka. Fördelen är även att det bli lätt att lägga till nya funktioner.

Produkterna är hårdkodade medans man själv skapar projekt som läggs till databasen.

Genom model Bind registreras ProduktId till respektive ProjektId. Med hjälp av filtrering kan därmed produkter till respektive projekt tas fram. Data används för att visas på View.

Reflektion

Nu börja MVC fastna och man känner sig hemma när man koda. Jag funderade först om det är upp till användaren att skapa sina egna produkter men tycker att detta inte är realistisk i ett "riktig" företag. Därför skapades "färdiga" produkter.

Ett av kraven som stod var att kunna korrigera prisuppgifter. Jag implementerade inte detta eftersom jag inte riktig förstod vem som skall ändra. Om det är upp till projektledare att ändra så kan priset vara väldigt lustigt. En Projektledare kanske tycker priset är rimligt men om en annan projektledare sedan ändra så kommer det påverka inte bara honom utan för allihopa. Därför togs inte en lösning för detta krav.

Ett annat krav är att kunna lägga till produkter till respektive projekt. Jag har tidigare gjort liknande upplägg med relationer (bageri 3.0, review) men tycker att den lösningen inte passa in här. Eftersom varje produkt hade isåfall hade varit kopplad till 1 projekt.

Med lite fundering och googlande kom jag fram till att jag förmodligen behöver ha en n:n databas som binder projektId med projektId. Här tog det dock stop eftersom jag inte lyckats få det att visa i view som jag ville. Jag tänker mig att man förmodligen i projekt skall direkt kunna komma åt alla produkter, men vet som sagt inte hur jag ska göra.

Därför fick jag använda filtrering av Bind som nödlösning och lämna in det eftersom det är deadline 😞. Borde finnas annan smidigare lösning utan filtrering?

Av: Fon To

Github: <https://github.com/fonto1/arkitekten.git>