Examensarbete

Django och React

Site för att knyta ihop försäljning med miljöinvesteringar

Maria Eidland - Webbutvecklare, Yrgo

Göteborg, 2019-11-14

Introduktion

Syfte

Målet med examensarbetet var att bygga en site som binder ihop klubbars försäljning med investeringar i miljöprojekt. Siten ska bestå av två delar där den ena delen innehåller en startsida öppen för alla besökare bestående av allmän information om konceptet - vad och varför, samt vilka miljöprojekt som finns att investera i. Från startsidan ska man även kunna logga in och skapa en ny användare. Den andra delen ska endast vara tillgänglig för inloggade användare och här ska man som försäljare kunna registrera beställningar av varor. I den delen ingår att registrera kunder och personuppgifter samt vilka varor de beställt.

Siten kommer byggas med JavaScript biblioteket React som frontend och ramverket Django som backend.

Bakgrund

Valet av tekniker som gjorts baseras på att det kan vara lärorikt att sätta sig in i både ett nytt ramverk och ett nytt programmeringsspråk då varken Django eller Python är något jag provat på tidigare. React är ett väldigt populärt JavaScript bibliotek som många siter är byggda med så det är bra att kunna ordentligt.

Frågeställningar

Två huvudsakliga frågeställningar som examensarbetet kommer avhandla är:

- Hur fungerar och används Django?
- Hur fungerar integrationen mellan Django och React?

Begränsningar

För att projektet inte ska växa och bli för stort har ett val gjorts om att inte ta med inloggning för klubbar. Det hade kunnat vara av intresse för dem att se statistik på

försäljning som gjorts av klubbens medlemmar samt vilka projekt de investerat i och hur utvecklingen av de projekten går. Ett val har även gjorts om att inte möjliggöra köp direkt på siten, det vill säga siten ska inte ses som en webshop utan som ett hjälpmedel för säljaren för att registrera beställning och säljarens uppgifter. Det skulle även kunna läggas till en del med vilka produkter som finns tillgängliga för försäljning men i grundversionen är det tänkt säljarna har dessa i en separat katalog.

Implementation

Django är ett ramverk för webbutveckling skrivet i Python. Det bygger på MVC konceptet, dvs. Model, View, Control samt med Serializers som konverterar data skickad i en HTTP förfrågan till ett Django objekt som gör om datan till t.ex. JSON.

Django Rest Framework fungerar som ett api som det går att hämta ut data från. Frontend och backend är därmed separerade från varandra men länkas ihop via anrop mot api:et. I Django är projektet uppbyggt av flera appar som hanterar olika delar av siten såsom användare, projekt och sitens huvudinnehåll.

Som nämnt ovan är Django skrivet i Pyhton. Syntaxen i Python är skapad för att vara enkel och konsekvent och bygger på konceptet att det bara ska finnas ett uppenbart sätt att skriva på. En skillnad från många andra programmeringsspråk är att det är kodens indentering som skapar block som kan ses i exemplet nedan av en funktion skriven i Python.

```
def foo(x):
    if x == 0:
        bar()
    else:
        foo(x - 1)
```

Vid deklarering av variabler anger man inte typ utan Python bygger på någon som kallas duck typing (eller automatic interfaces). “If it looks like a duck, swims like a duck and quacks like a duck then it probably is a duck”.

Resultat

Miljöerna för Django var enkla att sätta upp och det gick snabbt att komma igång med projektet trots att jag inte kodat i Python innan. För att kunna arbeta i Django krävs dock att man förstår hur hela ramverket hänger ihop så det var en del att sätta sig in i och lära sig. Så länge jag höll mig till enklare standardlösningar fungerade det relativt enkelt, men vid justeringar av dessa var det desto mer att sätta sig in i.

En svårighet med att få Django och React att fungera väl tillsammans var att skicka och ta emot data i rätt format. Vid vanlig fetch i React fungerade detta utan större problem men då jag bytte till att använda paketet Axios var det en hel del problem med fel dataformat.

Något som jag i efterhand borde ha använt är query sets som finns i Django för att göra förfrågningar till databasen. Jag hämtade istället ut all data från de aktuella tabellerna och använde mig av filter och map i React för att sedan få fram den data jag ville åt men det hade nog blivit tydligare och enklare att göra detta med query sets.

Ett problem jag stötte på var att jag till och från inte kunde logga in eller skapa nya användare, men upptäckte efter en del sökande att jag gjorde dubbla anrop till databasen vilket den inte kunde hantera. Detta resulterade i att data då och då inte postades.

Då det varit mer att sätta sig in i med Django och integrationen med React återstår en del funktionalitet på siten som jag ännu inte hunnit. Det som återstår är en profilsida där användaren kan redigera sina uppgifter och deleta användaren samt att kunna redigera uppgifter om köpare och ändra lagd order. Jag har inte heller hunnit sätta upp siten i produktion, men för att göra det kan man till exempel använda sig av Digital Ocean vilket det även finns guider för tillvägagångssätt vid användande av Django och React.

Generellt sett kan ändå konstateras att det fungerade relativt väl att arbeta med Django som backend och React som frontend. Django ger väldigt mycket på köpet så när man väl kan det och om man håller sig till enklare standardlösningar kan man sätta upp en fungerande site på kort tid.

Referenser

Django documentation: https://docs.djangoproject.com/

Django Rest Framework: http://www.tomchristie.com/rest-framework-2-docs/

Advantages and disadvantages with Django:
https://hackernoon.com/advantages-and-disadvantages-of-django-499b1e20a2c5

Python duck typing:
https://hackernoon.com/python-duck-typing-or-automatic-interfaces-73988ec9037f