

Bageri 3.0

Kodstruktur:

Strukturen för koden i nuläget är enligt MVC. För att uppfylla bagarens krav kommer ändringar att göras i Controller, Models samt View.

Nulägesrapport

Just nu har hemsidan en kontaktsida och en sida enbart för att lista produkter. Produkterna hämtas från en databas. Kodstrukturen är enligt MVC och implementerad enligt OOP.

Kravspecifikation

Produktkrav

- Besökare skall kunna lämna recensioner och omdömen på produkter
- Inloggningsfunktion skall implementeras, så bagaren vet vem som ger feedback
- Betygssystem 1–5 skall implementeras, medelvärde skall beräknas

Lösning/slutrapport

Inloggningsfunktion är framtagen mha ASP.net Identity. Användaren kan med hjälp av denna funktion skapa konto och logga in. Allting lagras i databasen.

Vidare har ProductController ändrats, en action-metod **Details** tar product-id som argument för att ge varje produkt en egen sida. Det är på denna sida en inloggad användare kan skriva kommentar samt lämna ett betyg.

Även metod **Save** är införd, denna har som uppgift att mha modelbinding hämta kommentar och betyg och på detta sätt lagra informationen i rätt attribut.

En ny model **Review** är införd och har som uppgift att spara dessa attribut.

För att lagra kommentarer och betyg har 1:n relation skapats mellan Product och Review.

Enhetstest

- Testet går ut på att testa om **Save** fungerar som det skall. 3 argument skickas till metoden därefter görs en kontroll på Review-count för att se om ny review är tillagt.

Reflektion

Betydligt lugnare vecka där det har varit gott om tid att hinna införa ändringarna samt hinna repetera gamla videos. Överlag har det varit rätt enkelt att slutföra uppgifterna.

Inloggningsfunktionen var väldigt lätt, medans **review** har mer varit en utmaning.

Från början skapade jag 2 attribut till review där jag tar indata med modelbinding. Med dessa visa jag sen informationen på **Details** sidan (branch bakery 3.0).

Lösningen fungerar men är inte lika bra som 1:n relation som jag tog fram vid ett senare tillfälle, eftersom data inte sparas i databasen lika fint som nu, där varje review är en egen rad (bakery 3.0b).

Utmaningen var att förstå vad som händer och hur man gör för att få det att funka (1: N relation).

Gill har hela tiden gjort detta med **Pie** och **Category** men aldrig förklarat eller gått in på detta vilket gör att jag själv bara följde efter men aldrig förstod vad som hände.

Detta är något jag tycker har varit genom hela kursens gång, dvs väldigt mycket copy paste efter honom men aldrig förstod vad och varför man gör så.

Av: Fon To

Github: <https://github.com/fonto1/Bakery>, branch bakery 3.0b