

# Webbutveckling 2: Stora projektet

Nu är det dags att tillämpa kunskaper från både Webbutveckling 1 och 2. I en stor grupp skall ni samarbeta och omvandla en gammal och "mindre snygg" webbplats till en modern och lättnavigerad produkt. Det skall finnas tydlig och tilltalande dokumentation, ett Drupal subtheme baserat på Bootstrap och såklart en färdig webbplats.

## Förutsättningar

Grupppuppgift.

## Resurser

- [Docker](#) tillsammans med filen [docker-compose.yml v.2.0](#)
- [Trello](#)
- [GitHub](#)
- [HTTrack](#)
- En editor som [Atom](#), [VSCodium](#) eller [Notepad++](#)

## Uppgiftsbeskrivning

### Ansvarsområden

I denna uppgift skall alla delta i att skriva kod, bidra med dokumentation och idéer. Men ni har också ett specifikt ansvarsområde som ni fått er tilldelade. Det innebär inte att ni endast skall göra det som ingår i ansvarsområdet, eller att endast ni skall göra uppgifter som faller under ansvarsområdet. Vad det innebär är att du bär ansvaret. Det är du som kommer få skulden om uppgiften inte utförs (oavsett om du är på lektionen eller inte).

Ansvarsområdena är *Arbetsledare*, *Gitmaster*, *Dokumentation*, *Data*, *Designsystem*, *Infrastruktur*.

Era *Arbetsledare* skall se till att ingenting hindrar arbetet. De skall ta ledningen under gruppdiskussioner och se till att det alltid finns tillgängliga arbetsuppgifter som kort på Trello.

De som agerar *Gitmaster* svarar på frågor och hjälper resten av gruppen med git-relaterade problem. De ansvarar också för att Pull Requests och git-konflikter löses.

Eftersom dokumentationen och rapportering av ert arbete är väldigt viktigt, bör alla vara inställda på att dokumentera ert arbete. Att detta arbete samordnas och sker kontinuerligt är en del av ansvarsområdet *Dokumentation*.

De gamla webbplatserna innehåller mycket information. Denna informationen är ofta dåligt sorterad och svår för besökare att hitta. Det krävs att de som är ansvariga för *Data* sätter sig in i hur man kan strukturera upp informationen och sortera den på ett logiskt och lättförståeligt sätt, både för kommande besökare samt användare av er webbplats. Den största uppgiften är dock att migrera data från den gamla webbplatsen till den nya.

Ett *Designsystem* kan börja livet som ett pattern library, men skall växa bortom det med både standarder för utseende och det "tänk" man skall ha när man bygger en webbplats. "Your product is more than just a pile of reusable UI elements. It has structure and meaning. It's not a generic web page, it's the embodiment of a system of concepts." skrev Emmet Connolly, director of product design på Intercom. Vidare skrev Diana Mounter, design systems manager på GitHub, "Design systems are always evolving, and the way you share and encourage adoption of new iterations will evolve along the way as well."

Det förekommer inte direkt något backend-jobb i detta projekt, men det finns en del *Infrastruktur* som behöver underhållas. Till exempel installation och konfiguration av moduler samt både testning och sanering av content types och exempeldata i Drupal.

### Regelbundna uppgifter

Börja varje lektion med en 5-10 minuter lång snabbgenomgång av hur arbetsläget ser ut. Låt var och en svara kort (c:a 1 mening) på frågorna "Vad gjorde jag förra gången?", "Vad skall jag göra idag?" och "Vilka problem ser jag?".

Varannan vecka skickar Dokumentationsansvariga in en rapport på It's där de kort redogör för vilka uppgifter som slutförts under de senaste fjorton dagarna, samt om det finns några problem som hindrar att gruppens arbete går vidare.

### **Större moment**

Uppdatera (eventuellt gör om) logotypen.

Bestäm färgschema, typografi och liknande.

Skapa ett design system.

Gör en sitemap. Basera denna på den gamla sidans innehåll, fast ta hänsyn till nya content types och taxonomier.

Tips! Ni behöver inte en "Hem"-länk eller en "Om oss"-sida. Denna informationen kan med fördel visas för besökaren genom att de först hamnar på en nyhetssida och att webbplatsen har en informativ footer.

Ta fram en Bootstrap 5 Theme-fil.

Bygg ett Drupal Bootstrap 5 (Barrio) subtheme.

Använd [YAML](#) och [Twig](#) för att skapa väl integrerade template-filer till webbplatsen. Förstasidan skall ha en egen template-fil, men det är upp till er vilka övriga sidor som behöver det.

Konfigurera Drupal med korrekta content types, taxonomier och liknande.

Migrera data från den gamla webbplatsen till er nya.

## Förväntat resultat

---

En layoutmässigt och grafiskt tilltalande dokumentation som beskriver färger, typsnitt, logotyper och andra grundkomponenter ni använder till sidan.

En .7z-fil som innehåller ett installeringsbart subtheme.

Komplett sida.

Alla skall ha bidragit till ert repository med vettiga commits, samt hanterat minst ett innehållsrikt kort på Trello i veckan.

## Vilka filer?

En 7z-fil med din kompletta subtheme-katalog så som den sett ut om den tankats ned från drupal.org.

Ett screenshot av förstasidan på din site.

## Var skall de lämnas in?

På vår lärplattform [It's Learning](#) finns ett kursrum med samma namn som den här kursen, under *Innehåll / Inlämningar* hittar ni en inlämningskatalog med samma namn som den här uppgiften.

---

Detta dokument är licenserat under Creative Commons BY-SA. Bilder och fotografier tillhör respektive upphovsman, och befinner sig inte nödvändigtvis under en Creative Commons-licens.

Dokumentet är skapat och uppdaterat av Kenneth Frantzen (@seetee på GitHub).