

# Labb 1

E-tjänster och webbprogrammering 7.5 hp VT-14

## Introduktion

I denna laboration kommer vi att arbeta med HTML, CSS, PHP, Javascript och MySQL. Målet är att bygga en kommentarssida. Slutresultatet (efter denna laboration) kommer alltså att vara en sida där vem som helst kan posta kommentarer, utan att ha en användare eller logga in. För att lösa denna labb behöver vi oundvikligen bekanta oss med `HTTP POST` och `HTTP GET`.

## Föreslagen struktur

Denna laboration består av 5 faser. Dessa är inte separata uppgifter, utan föreslagna steg för att bygga ovan nämnd applikation. Denna laboration kräver ingen inlämning. Men för att få en uppfattning om hur ett färdigt projekt efter denna labb skulle kunna se ut – beakta nedan filstruktur.

- labb1-fornamn-eternamn (mapp)
  - `index.php`
  - `assets` (mapp)
    - \* `img` (mapp med bilder)
    - \* `js` (mapp med javascriptfiler)
    - \* `css` (mapp med stylesheets)

Hur många filer ni har i varje mapp beror förstås på vilken strategi ni väljer att lösa problemet med. Läs vidare i instruktionerna för klarare förståelse.

# Uppgifter

Nedan följer uppgifterna som resulterar i inlämningen ovan.

## Uppgift 1 Formulär

Ett bra ställe att börja på kan vara att försöka få ett formulär att visas på en webbsida. Om du inte vill behöver du alltså inte involvera `php` än. Istället kan du arbeta med `html`.

Men poängen är alltså att – för att användaren ska kunna posta en kommentar krävs ett formulär. Vi bryr oss i denna övning inte om att försöka få formuläret att “fungera”. Börja helt enkelt med att försöka få ett `html`-formulär med relevanta fält att visas på en `html`-sida.

1. Börja med att skapa ett enkelt formulär bestående av minst 3 fält. Fälten som skall finnas är `namn` på den som postat inlägget, `själva inlägget` och `emailadress`. Sedan skall det förstås finnas en knapp för att kunna posta inlägget.

## Uppgift 2 Smyckning

I denna övning fokuserar vi på att sidans utseende. Alltså i huvudsak `css`. Skapa dig en mental bild (eller rita på ett papper) av hur du/ni skulle vilja att sidan ser ut. Försök sedan att realisera visionen så att sidan faktiskt ser ut som “mockup:en”.

Tänk på att ni under denna kurs kommer behöva kunna producera en webbapplikation med professionellt utseende. Så det finns **ingen anledning** att göra det för enkelt för sig.

- Styla er sida så att den inbjuder till att vilja posta kommentarer. Det skall klart och tydligt framgå vad sidan är till för. Det ska vara klart och tydligt hur man ska göra för att faktiskt posta en kommentar.
- Placera er `css` i en extern fil. Det är alltså inte tillåtet med `inline css`.
- Fundera över huruvida er sida är **responsiv** (som i **responsive design**). Hur skulle den kunna göras responsiv?

Oseriös stilning kommer ej att godtas.

## Uppgift 3 Validering – client-side

Denna övning går ut på att validera input. Ni skall se till att det finns en giltig e-postadress, att inga fält är tomma. Detta skall göras på klienten, alltså med `javascript`.

## Krav på validering

- Inga fält får vara tomma. Även en sträng med en massa mellanslag räknas som tom.
- E-postadressen måste vara en giltig e-post adress. Det räcker med att validera att den innehåller ett @-tecken, samt minst en punkt efter.

## Uppgift 4 Deigna databasen

I denna övning börjar vi närma oss en fungerande produkt. Vi börjar med att designa en databas.

- Skapa tabell som har hand om alla kommentarer.
- Tabellen måste ha en PK som genereras av databasen per inlägg. Se `auto increment`.
- Tabellen måste ha minst ett fält per fält i formuläret.

## Uppgift 5 Spara och visa data

Denna övning går ut på att faktiskt spara kommentarerna i en databas. Samt att visa dem. Ni väljer själva huruvida ni vill låta användaren posta kommentarer på samma sida som kommentarerna visas på. Eller om kommentarerna ska visas på en sida, och formuläret på en annan.

- Börja med att försöka få kommentarerna att läggas till i databasen när användaren postar formuläret. Verifera att inlägget faktiskt “kommer in” genom att kolla i databasen.
- När ni väl har inlägg i databasen. Försök då istället att skriva ut innehållet ifrån databasen på den sidan som ska visa kommentarerna. Ni väljer själva i vilken ordning kommentarerna ska visas på sidan.
- Style:a kommentarerna så att de passar den övriga layouten.

## Uppgift 6 Validering – server-side

Denna övning går ut på att validera datat som användaren skickar – men denna gång på server-sidan. Eftersom vemsomhelst kan “skapa” ett `HTTP request` utan att använda vårt

formulär, så spelar du ju ingen roll (rent säkerhetsmässigt) om vi validerar datat på klient-sidan eller inte. Vi validerar på klientsidan för att det ska vara trevligare för användaren. Vi validerar på server-sidan för att applikationen ska vara säkrare.

Applicera samma regler som vi gjorde på klient-sidan när du validerar server-sidan. Utöver dessa valideringskrav behöver även alla värden användaren konstruerat (och som ska in i databasen) skyddas genom `mysql_real_escape_string`.