

# Oblig 1

Laget av Rune Hjelsvold  
Høsten 2016

## Mål med øvingen

Bli kjent med MySQL/MariaDB og phpMyAdmin og få erfaring i å lage databaseapplikasjon på web med bruk av PHP og MySQL/MariaDB og PDO.

## Krav til innleveringen

Pass på at innleveringen tilfredsstiller disse kravene:

- **Format:** Fila dere laster opp på Fronter skal være i PDF. Koden legges ved som vedlegg i PDF-fila
- **Navnekonvensjon:** Fila dere laster opp på Fronter skal ha navnet *Oblig1\_etternavn\_studentnummer*, med ditt etternavn og studentnummer.
- **Programkode:** Programkode skal være ryddig og oversiktlig og skal følge konvensjonene i [http://www.tutorialspoint.com/php/php\\_coding\\_standard.htm](http://www.tutorialspoint.com/php/php_coding_standard.htm).

## Del I – Komme i gang med MySQL/MariaDB og phpMyAdmin

Denne oppgaven er til for å hjelpe dere å komme i gang med MySQL/MariaDB. Dere behøver ikke å lage noen rapport fra denne delen.

### Oppgave 1 – Skaffe tilgang til en database

Dere må først få opprettet en database dere kan bruke. Dere må gjerne installere MySQL/MariaDB og phpMyAdmin på egen maskin ([xampp](#) finnes for eksempel både for Windows, Linux og OS X). Dere kan også bruke MySQL på deres studentbruker (se [høgskolens informasjon om hjemmesider](#)) ved å fylle ut [en bestilling på våre nettsider](#).

### Oppgave 2 – Bli kjent med phpMyAdmin-brukergrensesnittet

Start nå opp en nettlester og gå til phpMyAdmin-siden på installasjonen (eller til [phpMyAdmin](#) på skolens server). Bruk litt tid til å gjøre det kjent med brukergrensesnittet og de ulike verktøyene og fanene. Det kan nå være lurt å besøke nettstedet til 24-7 Webs Inc. for å se på deres [phpMyAdmin Tutorials](#).

### Oppgave 3 – Importer en database

Jeg har laget en nedlastingsfil, Oblig1.zip, som dere kan hente ned fra Fronter. Her finner dere bl.a. en liten applikasjon som gjør bruk av en database som inneholder informasjon om en håndfull CD-er. SQL-skriptet CreateCdLibrary.sql inneholder SQL-setninger som vil generere denne databasen. Importer databasen ved bruk av phpMyAdmin sin importfunksjon. Fila er lagret i UTF-8-format. Velg derfor UTF-8 i nedtrekksmenyen for "Character set of the file".

Gjør dere kjent med strukturen og innholdet i denne databasen.

## Oppgave 4 – Installer demoapplikasjon

Eksempelapplikasjonen er skrevet i PHP og gjør bruk av databasen dere nettopp har importert. Med denne applikasjonen kan dere liste ut informasjon om CD-ene i arkivet og dere kan legge inn informasjon om flere CD-er. Applikasjonen finnes i to versjoner:

- **Plain:** I denne versjonen er programkode for å aksessere database kombinert med kode for selve applikasjonslogikk og for brukergrensesnittet.
- **MVC:** Denne versjonen demonstrerer hvordan man kan gjøre bruk av Model-View-Controller-mønsteret i utvikling av databaseapplikasjoner på web.

Gjør dere kjent med begge versjonene.

## Del II – Komme i gang med PHP

Denne oppgaven er også til for å hjelpe dere å komme i gang med PHP. Dere behøver ikke å lage noen rapport fra denne delen heller.

### Oppgave 1 – Bli kjent med HTML

Jobb gjennom W3C sin [HTML/Training wiki](#). Dere kan stoppe etter uke 3.

### Oppgave 2 – Bli kjent med PHP

Jobb gjennom tutorialpoint sin [PHP Tutorial](#). Dere kan hoppe over kapitlene «PHP - Cookies», «PHP - Sessions» og «PHP - Sending Emails». Fra seksjonen «Advanced PHP» trenger dere bare å se på «PHP - Predefined Variables», «PHP - Error Handling», «PHP - Bugs Debugging», «PHP - Date & Time», «PHP - Object Oriented» og «PHP - For C Developers».

### Oppgave 3 – Bli kjent med MVC

Jobb gjennom php-html.net sin [Model View Controller \(MVC\) in PHP](#).

### Oppgave 4 – Bli kjent med PDO

Vi vil bruke PDO som API mot MySQL i dette kurset. Dere finner mange eksempler på nettet der MySQL sin egen API er brukt. Ikke kopier dette! Jobb gjennom hashphp.org sin [PDO-tutorial for MySQL](#) - det vil gi dere en god introduksjon til PDO.

## Del III – Lage databaseapplikasjon

I denne delen av oppgaven skal dere lage deres egen webapplikasjon ved bruk av PHP, PDO og MySQL. Oppgaven går ut på å lage en blog der enhver nettbruker kan legge inn et innlegg, og der alle nettbrukere kan se alle innlegg som er lagt inn. Til denne oppgaven skal dere lage en rapport der dere kort beskriver databasen og applikasjonen dere har laget. Dere skal legge ved databasen eksportert til SQL-fil (med noen få eksempelinlegg) og dere skal legge ved all PHP-kode dere har laget. Dere kan velge om dere ønsker å bruke MVC-mønsteret for applikasjonen.

### Oppgave 1 – Lag database for blogapplikasjonen

Opprett en tabell for bloginnlegg i din database der du definerer disse kolonnene:

- **Id:** Alle bloginnlegg skal ha unik id. Id kan gjerne være av typen INT. Bruk gjerne `auto_increment` for generering av id-er.
- **Tittel:** Alle bloginnlegg skal ha en tittel. Tittelen kan gjerne være av typen VARCHAR på inntil 255 tegn.
- **Innhold:** Innholdet i innlegget kan gjerne være av typen TEXT.
- **ForfatterNavn:** Navnet på den som har skrevet innlegget kan gjerne være av typen VARCHAR på inntil 255 tegn.
- **ForfatterEpost:** E-postadressen på den som har skrevet innlegget, kan gjerne være av typen VARCHAR på inntil 255 tegn.
- **PubTime:** Dato da innlegget ble publisert (dvs. skrevet) kan gjerne være av typen DATETIME eller av typen TIMESTAMP. Bruk gjerne MySQL-funksjonen `NOW()` for å generere tidspunktet automatisk – eventuelt kan dere sette up databasen til å bruke `CURRENT_TIMESTAMP` som default verdi.

Legg inn et par rader i blogtabellen ved bruk av phpMyAdmin slik at dere har noe data for testing av PHP-koden dere skal lage i neste oppgave.

### Oppgave 2 – Lag PHP-kode for å vise bloginnleggene

Dere skal i denne oppgaven lage PHP-kode for å hente ut alle bloginnlegg fra databasen ved bruk av PDO og vise dem fram i en nettleser. Bloginnlegg kan gjerne ha denne HTML-koden (se også eksempelfila `BlogInnhold.html`):

```
<h2>Innleggets tittel</h2>
<p>
<em>Forfatterens navn (forfatterens e-post)</em>
</p>
<p>
Selve innlegget
</p>
<p>
<em>Tid innlegget ble skrevet</em>
</p>
```

Krav til programkoden:

- Dere må bruke `htmlspecialchars` i PHP når genererer HTML for data hentet fra databasen for å unngå at spesielle tegn lagret i databasen (spesielt tegnene `<` og `&`) gir HTML-problemer.
- Dere skal presentere blogginnleggene med de nyeste innleggene først. (Tips: dere kan hente ut blogginnleggene sortert med nyeste innlegg først ved å legge til `ORDER BY PubTime DESC` i SQL-setningen dere bruker mot databasen.)
- Dere skal navngi alle kolonnene dere benytter i SQL-setningen – dvs. vi vil ikke akseptere `SELECT *` i SQL-setningen dere bruker.
- Dere skal lage ei egen php-fil der databasenavn og passord ligger slik at påloggingsinformasjon ikke er blandet med applikasjonskode og slik at påloggingsinformasjon kan importeres inn – og ikke dupliseres - i de ulike skriptene.

### Oppgave 3 – Lag PHP-kode for å legge inn et innlegg i bloggen

Dere skal i denne oppgaven lage PHP-kode for å legge inn nye innlegg i bloggen. Dere må da lage en HTML form der brukeren kan skrive inn tittel og innhold i innlegget i tillegg til sitt eget navn og epost-adresse. Dere kan gjerne bruke det enkle designet i eksempelfila `BlogForm.html` som utgangspunkt for PHP-koden. Her er et utdrag fra denne fila:

```
<form action='addBlogEntry.php' method='post'>
Innleggets tittel:
    <input type='text' name='tittel' size='42' /><br>
Innholdet i innlegget:<br>
<textarea rows='8' cols='50' name='innhold'></textarea><br>
Ditt navn: <input type='text' name='navn' size='42' /><br>
Din e-postadresse:
    <input type='text' name='epost' size='42' /><br>
<input type = 'submit' name = 'Send innlegg' />
</form>
```

Krav til programkoden:

- Dere **skal** bruke PDO (ikke mysql- eller mysqli-bibliotekene).
- Dere skal avvise blog-innlegg dersom ikke brukeren har fylt inn alle feltene i formen.
- Dere skal håndtere mulige databasefeil (f.eks. skal ikke koden kræsje men gi en fornuftig feilmelding dersom databasen av en eller annen grunn er nede).
- Dere må bruke ***prepared statements*** i PDO for å unngå at sluttbrukeren forsøker å legge inn data som skaper problemer for systemet deres.