



Webbutveckling med PHP

# PROJEKT, INKLUDERA FILER

Utbildare: Mikael Olsson

[mikael.olsson@emmio.se](mailto:mikael.olsson@emmio.se)

076-174 90 43

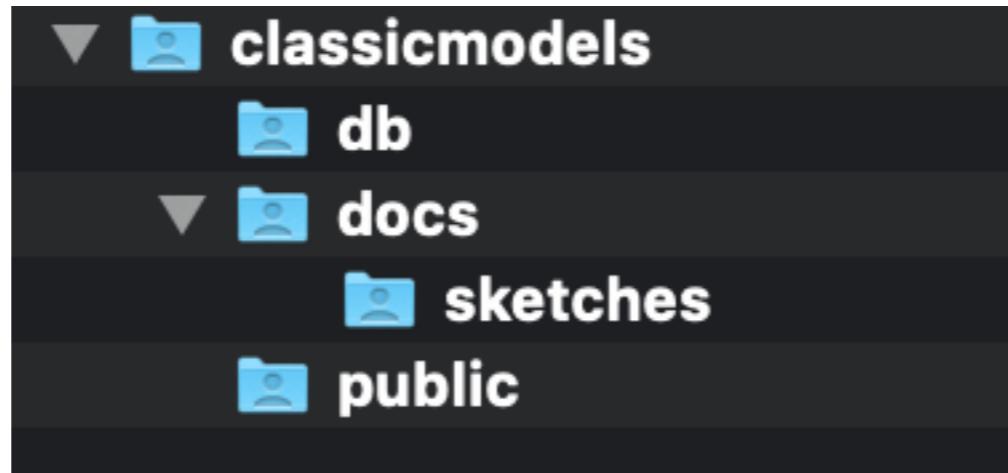
# NACKADEMIN

# Starta ett projekt

- Projektmapp
- Host (local/live)
- Repo
- IDE (kod/db)
- Backlog
- ERM
- DoD
- Socialt kontrakt
- Set expectations / planera
  - Sprint planning
- Jobba ihop, dra nytta av varandras kunskap
  - Daily scrum
- Inspektera increment
  - Sprint review
- Utvärdera arbetet
  - Sprint retrospective

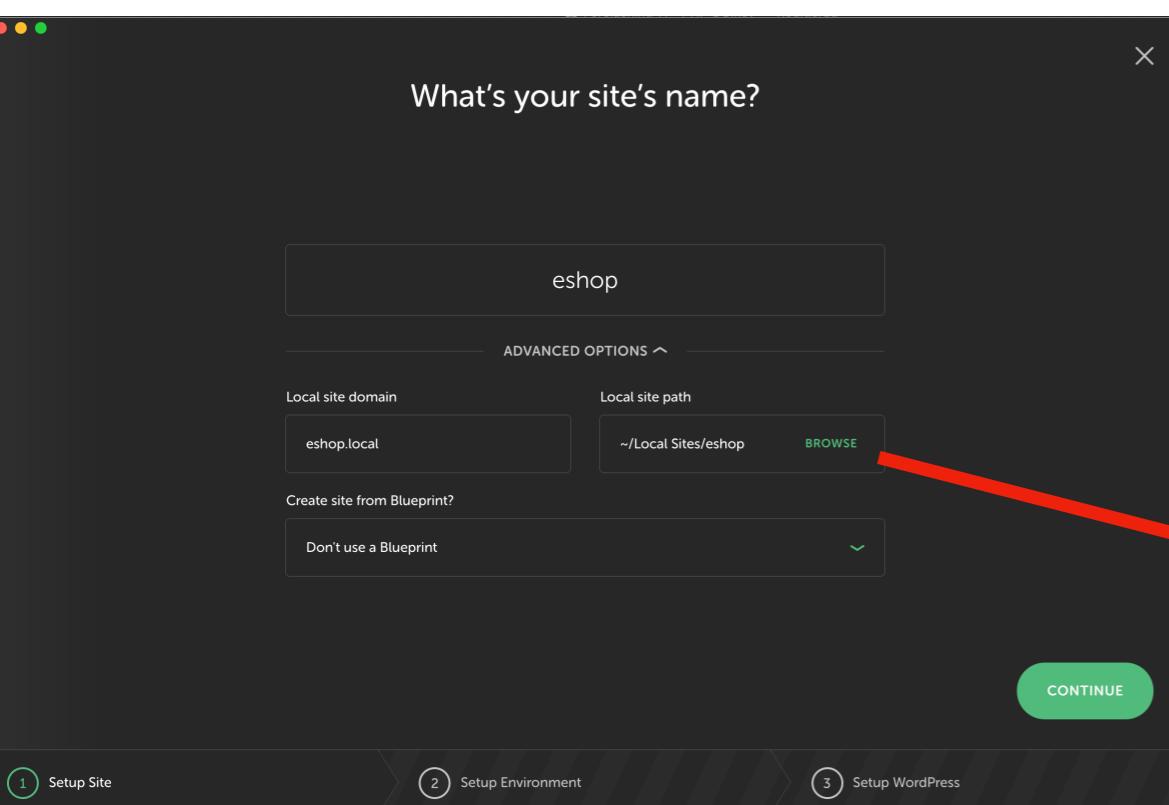
# Projektmapp

- Man kan strukturera upp sin projektmapp på olika sätt.
  - Man kanske vill samla olika delar som hör till projektet.

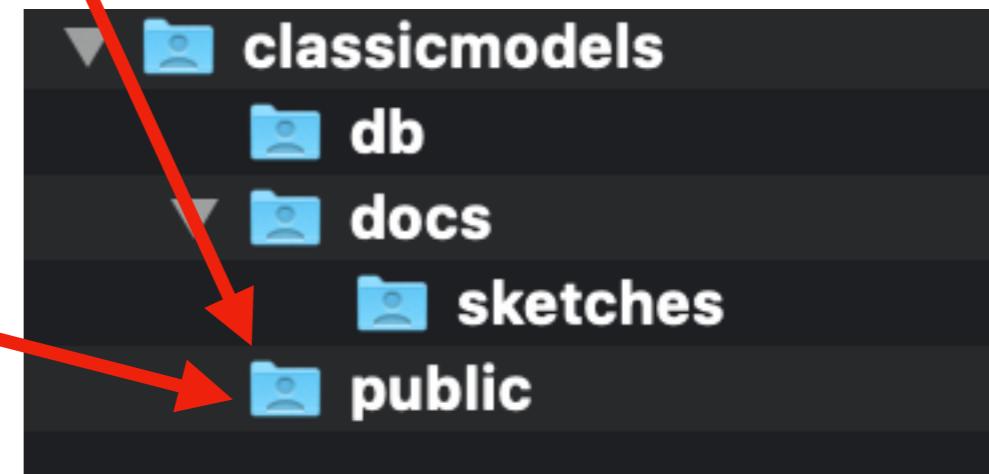


# Host - local

- I apache eller nginx sätter man upp en VirtualHost där vi kan ange en DocumentRoot, dvs var filerna för projektet ska finnas.
- Local by Flywheel sköter mycket åt oss så vi inte behöver grotta ner oss i det tekniska.



```
<VirtualHost *:80>
  DocumentRoot "/var/www/public/"
  ServerName example.com
  DirectoryIndex index.html index.php
</VirtualHost>
```



# Host - public

- Webhotell
  - Leverantörer
    - Loopia
    - Binero
    - one.com
    - Oderland
  - Kräver oftast en domän
- Plus
  - Minimalt underhåll och teknisk kunskap
- Minus
  - Inte så stora anpassningsmöjligheter om man t ex behöver en specifik version av PHP, nån speciell extension eller annat program.

# Host - public

- Andra lösningar
  - VPS
  - Managed hosting
  - Egen dator
- Plus
  - Större frihet för egna program t ex
- Minus
  - Kräver större ansvarstagande

# Flera hostar

- Man har ofta flera publika eller "halvpublika" hostar, t ex en host som användarna kommer åt (produktion, live) och en gemensam testserver (dev, test, stage) där man (utvecklare, testare, produktägare) kan testa att funktionaliteten fungerar tillsammans med all annan kod innan man släpper den.
- Vem bestämmer (i Scrum) om funktionalitet ska släppas?

# Repo

- Enklast att sätta upp repot remote först och sedan klona.
- <https://github.com/new>
- Mappen (public i det här fallet) måste vara tom.

## Create a new repository

A repository contains all project files, including the revision history.

Owner Repository name \*



emmio-micke

/ classicmodels



Great repository names are short and memorable. Need inspiration? How about [improved-carnival?](#)

Description (optional)

Public

Anyone can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None



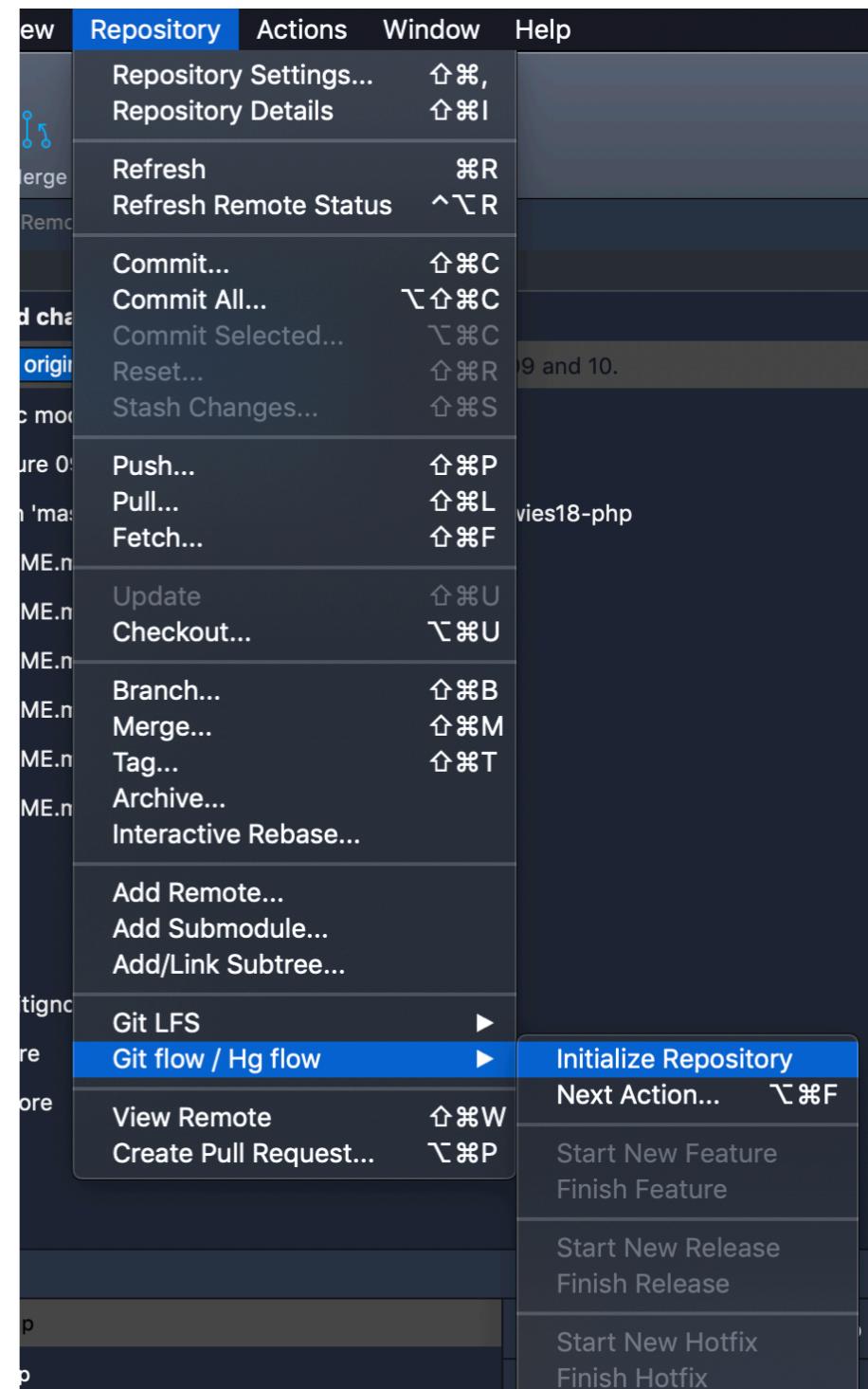
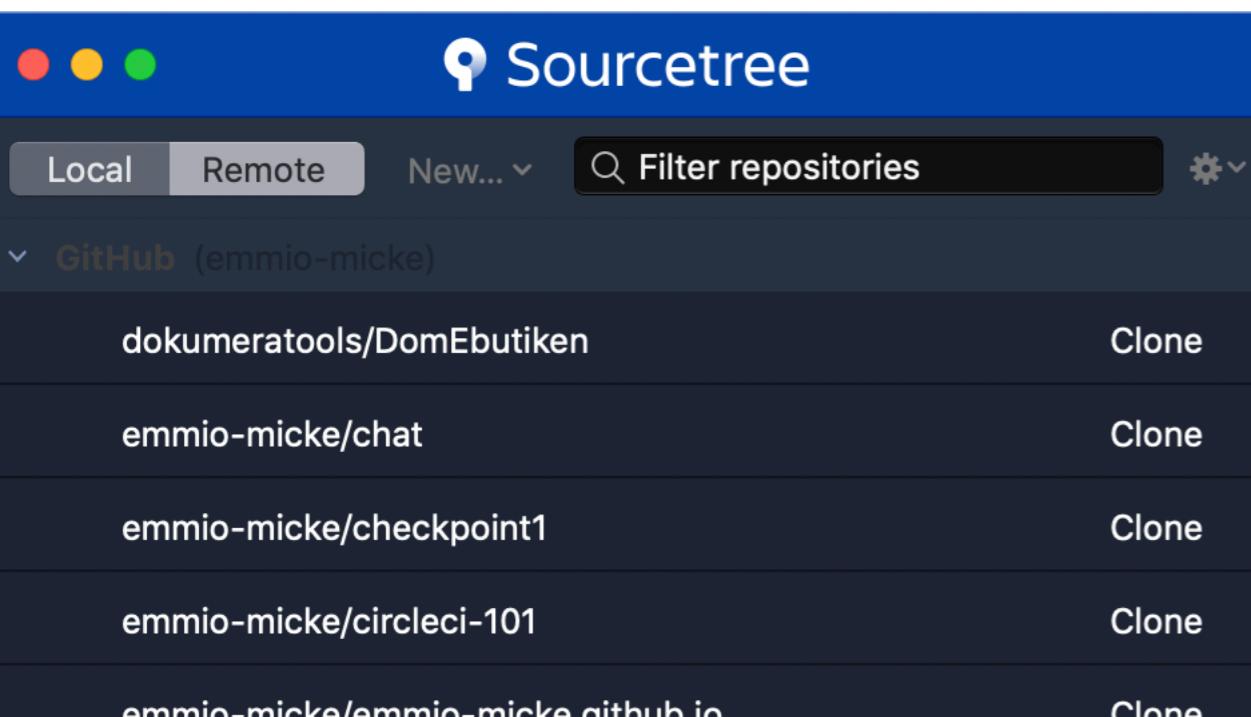
**Create repository**

```
Mikael's-MacBook-Pro:classicmodels micke$ ls -la
total 8
drwxr-xr-x@ 7 micke  staff  224 25 Feb 14:02 .
drwxr-xr-x@ 6 micke  staff  192 25 Feb 08:28 ..
drwxr-xr-x@ 14 micke  staff  448 25 Feb 14:02 .git
-rw-r--r--  1 micke  staff    5 25 Feb 12:58 .gitignore
drwxr-xr-x@ 3 micke  staff   96 17 Feb 16:26 db
drwxr-xr-x@ 4 micke  staff  128 17 Feb 16:26 docs
drwxr-xr-x@ 9 micke  staff  288 28 Feb 14:17 public
```

```
Mikael's-MacBook-Pro:classicmodels micke$ git clone https://github.com/emmio-micke/classicmodels.git public
```

# Repo

- Sourcetree har stöd för att klona remote repon och för Git flow.



# Repo

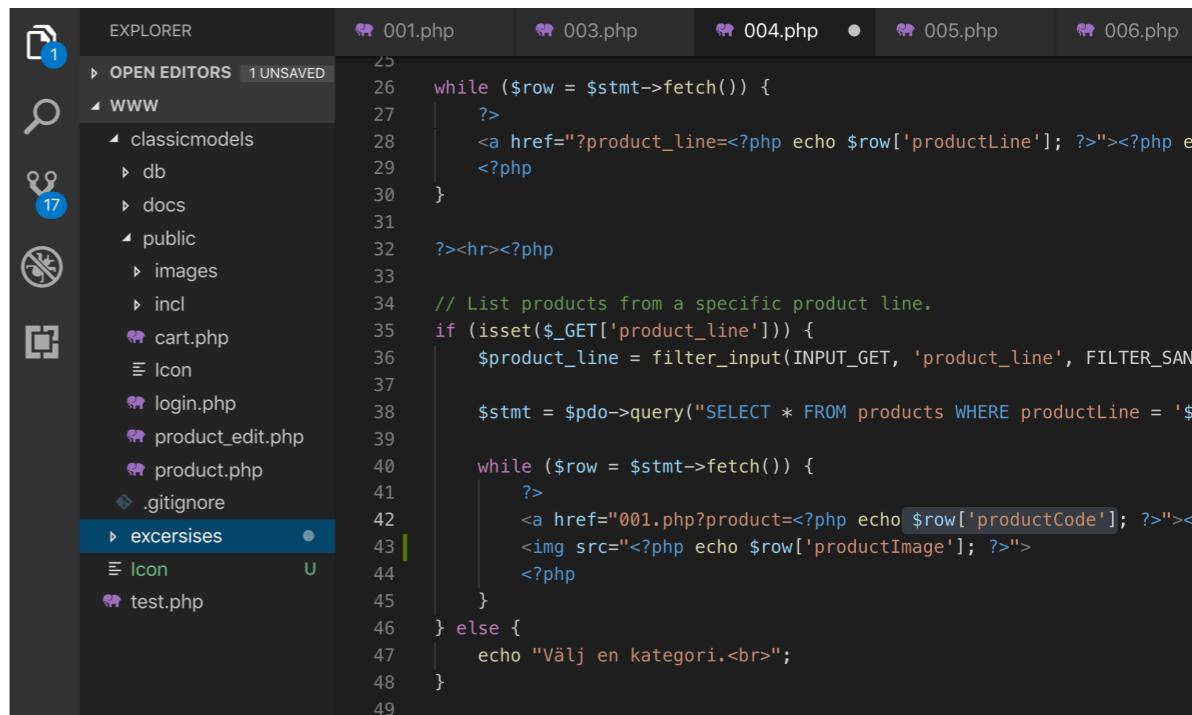
- Det finns olika lösningar / extensions för VSC om man hellre kör det, t ex:
  - [https://marketplace.visualstudio.com/items?  
itemName=GreatMinds.gitflow4code](https://marketplace.visualstudio.com/items?itemName=GreatMinds.gitflow4code)

```
>gitflow|
```

GitFlow: Finish Feature	⌘ ⌘ / ⌘ F
GitFlow: Finish Hotfix	⌘ ⌘ / ⌘ H
GitFlow: Finish Release	⌘ ⌘ / ⌘ R
GitFlow: Git Status	⌘ ⌘ / ⌘ S
GitFlow: Initialize GitFlow	⌘ ⌘ / ⌘ I
GitFlow: Start a Feature	⌘ ⌘ / ⌘ F
GitFlow: Start a Hotfix	⌘ ⌘ / ⌘ H
GitFlow: Start a Release	⌘ ⌘ / ⌘ R

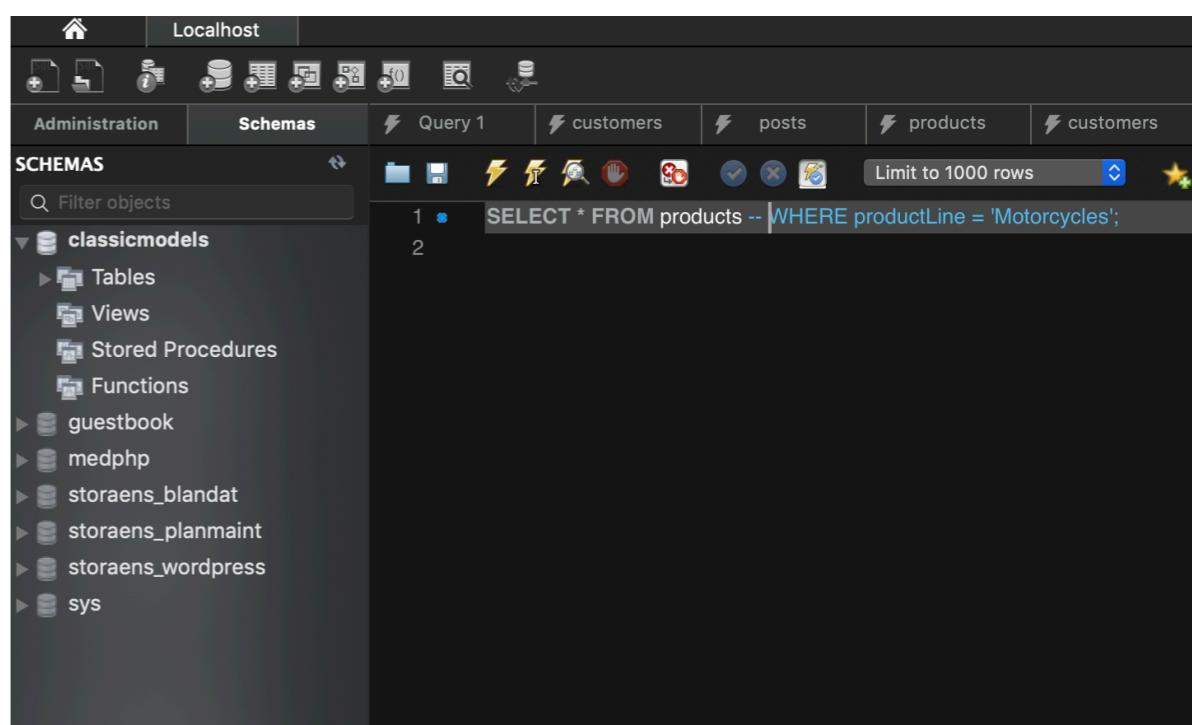
# IDE

- Dra din public-mapp till VSC.
- Sätt upp Workbench.



A screenshot of a code editor interface. On the left is the Explorer sidebar showing a file tree with folders like 'classicmodels', 'public', and 'excercises'. The main area displays a PHP script with line numbers from 25 to 49. The code includes comments and logic for listing products from a specific product line.

```
25 while ($row = $stmt->fetch()) {  
26     ?>  
27     <a href="?product_line=<?php echo $row['productLine']; ?><?php e  
28     <?php  
29 }  
30 ?><hr><?php  
31 // List products from a specific product line.  
32 if (isset($_GET['product_line'])) {  
33     $product_line = filter_input(INPUT_GET, 'product_line', FILTER_SANITIZE  
34     $stmt = $pdo->query("SELECT * FROM products WHERE productLine = '$  
35     while ($row = $stmt->fetch()) {  
36         ?>  
37         <a href="001.php?product=<?php echo $row['productCode']; ?><?php e  
38             <img src=<?php echo $row['productImage']; ?><?php  
39     }  
40 } else {  
41     echo "Välj en kategori.<br>";  
42 }  
43  
44  
45  
46  
47  
48  
49 }
```



A screenshot of MySQL Workbench. The top navigation bar shows 'localhost'. The left pane displays the database schema with 'classicmodels' selected, showing tables, views, stored procedures, and functions. The right pane shows a query editor with a single result row for a SELECT query.

Administration      Schemas

SCHEMAS

classicmodels

- Tables
- Views
- Stored Procedures
- Functions

guestbook  
medphp  
storaens\_blandat  
storaens\_planmaint  
storaens\_wordpress  
sys

Query 1    customers    posts    products    customers

1    SELECT \* FROM products -- WHERE productLine = 'Motorcycles';  
2

# Backlog

- Oavsett vilket system man använder behöver man sätta upp nån slags struktur.

The image shows a digital backlog management interface with the following columns:

- Ideas**: Contains items like "Implement migration for issues with DB alteration" and "Speed up listings".
- Product backlog**: Shows a "Star graph" diagram with nodes for Governance, Environment, Human Rights, Labour rights, and Anticipation.
- Sprint backlog**: Displays a card for a bug titled "Failed to load resource: the server responded with a status of 500 ()". It also shows an "Inspection Protocol" card with the message "It doesn't seem to be possible to save an IP".
- In progress**: A placeholder card with the text "+ Lägg till ett kort".
- Ready for testing**: Shows a "Corrective Action Plan (CAP)" document and a "Supplier View ACAP" section.
- Back fr**: A partially visible column on the right.

# Detalj - namn

- Vem är med i vilken trello-tavla?



# Att göra ett kort

- I just detta projekt har ni ett par kort som ni är ansvariga för, beroende på ambitionsnivå, men vanligen plockar man det högst prioriterade kortet.
- Vem prioriterar korten?

The image shows a mobile application interface with a light gray background and a blue header bar. The header bar contains the text 'Sprint backlog' on the left and three vertical dots on the right. Below the header, there are five horizontal cards, each with a blue rounded rectangle at the top containing a category name. The first card is labeled 'App' and contains the text: 'We should have a more clear way to answer Yes/No/NA questions, maybe with another kind of component.' followed by a three-line menu icon. The second card is labeled 'Inspection Protocol' and contains the text: 'Simplify by removing some field options' followed by a three-line menu icon. The third card is labeled 'Inspection Protocol' and contains the text: 'We need an inspector role' followed by a three-line menu icon. The fourth card is labeled 'Inspection Protocol' and contains the text: 'Ability to assign inspector to Inspection Protocol' followed by a three-line menu icon. The fifth card is labeled 'App' and contains the text: 'Inspector should be able to submit a' followed by a three-line menu icon.

Sprint backlog

App

We should have a more clear way to answer Yes/No/NA questions, maybe with another kind of component.

Inspection Protocol

Simplify by removing some field options

Inspection Protocol

We need an inspector role

Inspection Protocol

Ability to assign inspector to Inspection Protocol

App

Inspector should be able to submit a

# Att göra ett kort

- Vi kan jobba med medlemmar, så vi ser vem som gör vilket kort.
- Vi kan jobba med etiketter, t ex för olika delar av siten (ordrar, produkter osv), olika typer av kort (feature requests, buggar osv), vi kan bifoga filer, länkar, screenshots, skisser osv.

The screenshot shows a Scrumcoo card interface for a 'Development issue' card. The card has the following sections:

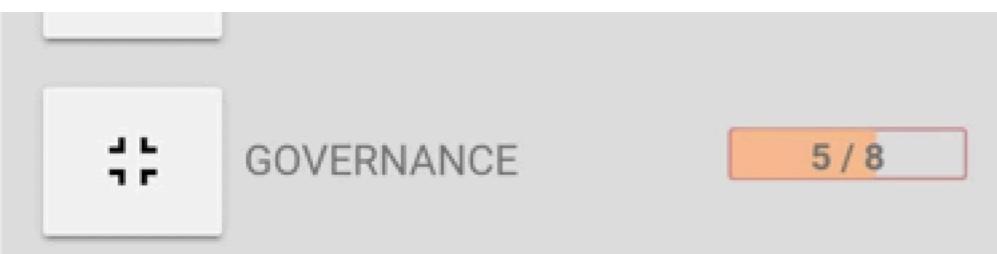
- Development issue**: Status: In progress. Members: Mikael Olsson. Labels: Development, Bug.
- Beskrivning**: Placeholder: Lägg till en mer detaljerad beskrivning...
- Lägg till kommentar**: Placeholder: Skriv en kommentar... with a rich text editor toolbar.
- Aktivitet**:
  - Mikael Olsson gick med i detta kort (3 nov 2018 15:21)
  - Mikael Olsson la till detta kort i In progress (3 nov 2018 15:20)

On the right side, there are several sidebar buttons:

- SCRUMCOON: Estimate in SC
- LÄGG TILL PÅ KORT: Medlemmar, Etiketter, Checklista, Förfallodatum, Bilaga
- POWER-UPS: Skaffa Power-Ups
- ÅTGÄRDER: Start timer, Flytta, Kopiera, Bevaka (marked with a checkmark)
- Arkivera, Dela

# Att göra ett kort

- Bilder kan göra det tydligare att se vad vi förväntar oss av kortet, både för buggar och ny funktionalitet.



- Does the company p..
- Does the Code inclu...
- Is the Code posted i...
- Are workers informe...

Göm detaljer

### Aktivitet

Mikael Olsson bifogade [image.png](#) till detta kort Igår 10:45

Element.attachShadow instead. See <https://www.chromestatus.com/features/4507242028072960> for more details.

3 Failed to load resource: the server responded with a status of 500 ()

Svara

Mikael Olsson bifogade [image.png](#) till detta kort Igår 10:45

ESAT

ERROR:

OK

Svara

Mikael Olsson bifogade [image.png](#) till detta kort Igår 10:45

There are unsaved changes →

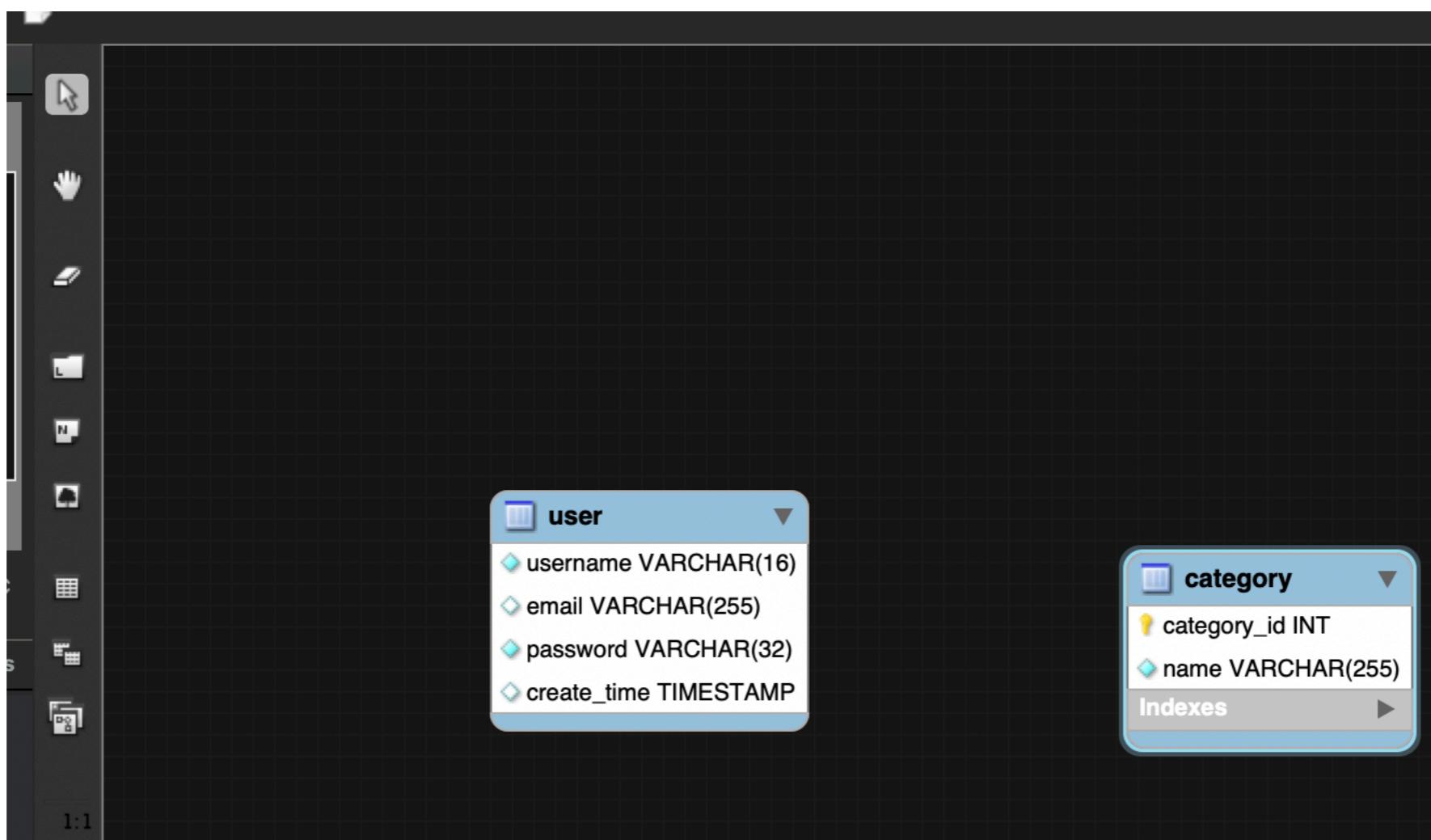
Download icon

Save icon

Mandatory Off

# ERM

- Man kan dra upp riktlinjerna i början av projektet men överarbeta inte, ni kommer ändå att behöva ändra modellen längs vägen. (Late decisions.)



# Definition of Done

- En DoD är en överenskommelse om vad det innebär att kalla något för klart.
- En DoD ska appliceras på varje kort, inte på hela projektet.
- Man kan ha olika DoD för olika delar av projektet, t ex Ready for testing, Release osv.
- När uppdaterar man en DoD?
  - Koden fungerar enligt spec.
  - Koden är incheckad.
  - Testerna fungerar.
  - Koden är peer reviewad.
  - Dokumentationen är uppdaterad.
  - Det går att testköra funktionaliteten.

# Socialt kontrakt

- Vissa team väljer att skapa sociala kontrakt för att formalisera hur teamet ska fungera.
  - Hur ställer vi oss till distansarbete?
  - Hur balanserar vi att hjälpa varandra och få egen tid för att koda?
  - Är det okej att äta mat vid arbetsplatsen?
  - Hur ger vi varandra feedback?
  - Är det okej att inte vara med på daily?
- När uppdaterar man kontraktet?
- <http://www.ethann.com/social-contracts-for-agile-teams/>

# Grupparbeten

- Det ingår i ett grupparbete att ni ska kunna arbeta ihop.
- Ta upp problem tidigt. Först och främst i gruppen, i andra hand med mig.
- Hur kan ni definiera att arbetet går framåt? Hur främjar vi ärlighet på daily? Hur ser vi till att kod integreras ofta?

# Arbetsflöde

- Scrum som exempel, men huvudsaken är att hitta ett flöde som fungerar och gör oss produktiva.
  - Sprint planning
  - Daily Scrum
  - Sprint review
  - Sprint retrospective

# Sprint planning

- Gå igenom product backlog tillsammans med kravställare (product owner). Förstår alla vad som ska göras i varje kort? Komplexitet/tid?
- Åta er vad ni ska hinna med. (Svårt att uppskatta tid med föreläsningar osv.)

# Daily Scrum

- Status
  - Vad har jag gjort sen igår?
  - Vad ska jag göra idag?
  - Är det något som hindrar mig?
- Fastna inte i lösningsmöten.

# Sprint review

- Inspektera / demonstrera vad ni har gjort.
- Samla in feedback från stakeholders.
- Justera product backlog.

# Sprint retrospective

- Vad har fungerat bra under sprinten?
  - Fungerar teamet bra? Har ni löst ett svårt problem? Någon som varit extra hjälpsam?
- Vad kan vi förbättra?
  - Lära oss mer om någon teknik? Ramverk? Kommentera kod annorlunda? Sitta ihop i andra konstellationer? Andra fikatider? Fungerar lokalen bra?
- Viktigt att inte leta syndabockar.
  - Safe place. Accepterande, prestigefri miljö. Tänk mer "process" än "person".
- Det kan vara svårt att komma på saker på mötet. Skriv gärna upp saker under sprinten när ni kommer på dem.
- Sträva efter att göra nästa sprint lite bättre än den föregående.

# Dagligt arbete

- Välj en prioriterad uppgift.
- Vet du hur du ska lösa den? Vet någon i ditt team? (Utveckling, affärskunskap mm.)
- Git flow: Skapa en feature-branch.
  - När ska nimerga tillbaka till dev?
  - Vem bestämmer om en feature är leveransklar eller inte i Scrum?
  - Om featuren inte är klar att merga; byt branch och börja på nästa uppgift.
- Behöver ni skisser? Diskutera lösningar med teamet?
- Parprogrammera gärna emellanåt. Det hjälper er när ni har svårt att komma igång och främjar kunskapsdelning.

# Peer review

- Vad ska man titta efter?
  - Är koden kommenterad? Lätt att förstå?
  - Löser koden problemet? Finns det logiska fel?
  - Effektiv? Borde man refaktorera någon kod?
  - Följer ni någon kodstandard?

# Jobba i projekt

- Saknar ni något eller vet ni hur ni ska fortsätta jobba?

# Inlämningsuppgifter

- Vi kommer att behöva ha en liten individuell inlämningsuppgift, en lite större individuell inlämningsuppgift och en gruppuppgift.
- Jag tror inte att det går att göra två betygsgrundande projekt i en vecka med fyra föreläsningar.
- Det viktiga är inte mängden funktioner utan att ni visar att ni har förstått innehållet.
- Förslag:
  - Gruppuppgift vecka 3 som innehåller en klass som hanterar inloggning via AJAX. G eller IG. Kan innehålla uppgifter som individer kan lösa för VG.
  - Individuell uppgift vecka 4 som innehåller API, Composer och Dependency injection. Generalisering, **v4** deploy.

Lektionstillfälle:	Centralt innehåll:
mån 13 jan.	Grundläggande Webb, Apache
tis 14 jan.	Grundläggande PHP
tors 16 jan.	Jobba i projekt, inkludera filer, cookies/sessions
mån 20 jan.	Objektorientering
tis 21 jan.	Felsökning / tester, hantera formulär, scope
ons 22 jan.	DB, MySQL, PDO, lösenord
tors 23 jan.	Mer DB, Frontend & PHP
<b>mån 27 jan.</b>	<b>API</b>
<b>tis 28 jan.</b>	<b>Generalisering</b>
<b>ons 29 jan.</b>	<b>Namespace / felhantering / Composer / mail</b>
<b>tors 30 jan.</b>	<b>Dependency Injection</b>
mån 3 feb.	Ev Laravel / Eloquent
tis 4 feb.	
ons 5 feb.	Nytt i PHP7, Webservrar / säkerhet, rättigheter
tors 6 feb.	Redovisning enskilt

# Inlämningsuppgift 1

- Utgå ifrån denna sida:  
<http://sandbox.onlinephpfunctions.com/code/085855c86f39d201ee87fe60a719802a2b667e9c>
- Ändra koden så att skriptet skriver ut vilket stjärntecken datumet skulle ge.  
<https://sv.wikipedia.org/wiki/Stj%C3%A4rntecken>
- Du måste jämföra datum, du får inte jämföra strängar.
- Lämna in genom att skapa en share-länk och skicka in den senast söndag 19/1. [Save or share your code](#)
- Uppgiften kan ge G eller IG.

**Result:**

En person född 1998-03-01 skulle ha stjärntecknet Fiskarna.

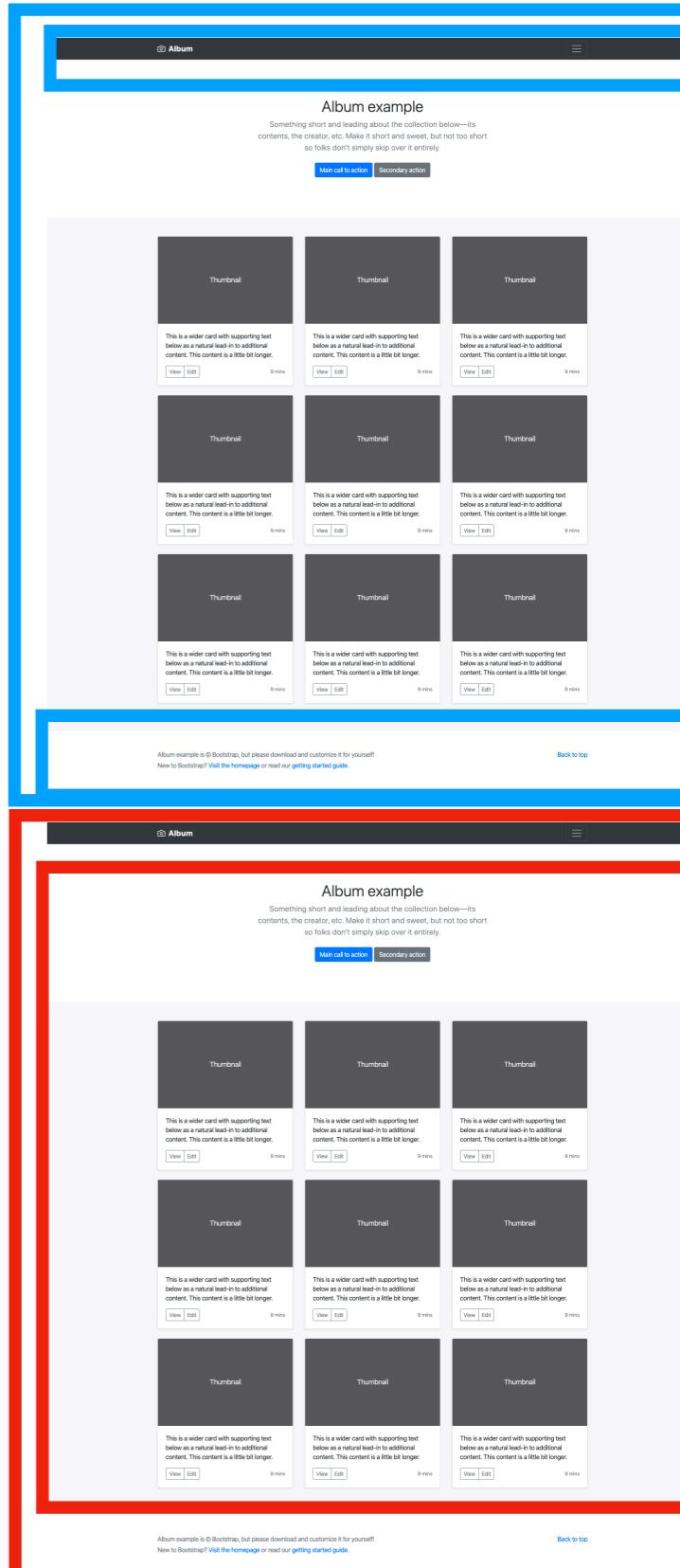
Lektionstillfälle:	Centralt innehåll:
mån 13 jan.	Grundläggande Webb, Apache
tis 14 jan.	Grundläggande PHP
tors 16 jan.	Jobba i projekt, inkludera filer, cookies/sessions
mån 20 jan.	Objektorientering
tis 21 jan.	Felsökning / tester, hantera formulär, scope
ons 22 jan.	DB, MySQL, PDO, lösenord
tors 23 jan.	Mer DB, Frontend & PHP
mån 27 jan.	API
tis 28 jan.	Generalisering
ons 29 jan.	Namespace / felhantering / Composer / mail
tors 30 jan.	Dependency Injection
mån 3 feb.	Ev Laravel / Eloquent
tis 4 feb.	
ons 5 feb.	Nytt i PHP7, Webservrar / säkerhet, rättigheter
tors 6 feb.	Redovisning enskilt

# Inkludera filer

- include - include\_once
  - Försöker inkludera filen men fortsätter skriptet om det inte går.
  - <http://php.net/manual/en/function.include.php>
- require - require\_once
  - Försöker inkludera filen och avbryter skriptet om det inte går.
  - <http://php.net/manual/en/function.require.php>

# Övning

- Skapa filen `header.php` som innehåller en html-header och `footer.php` som innehåller en footer.
- Skapa filen `page1.php` som bara innehåller body-delen och inkluderar headern och footern.
- Extra: Skapa en variabel `$title` i filen `page1.php` och använd den för att sätta titeln på sidan i `header.php`.
- Extra: Nytt projekt, gör ”tvärtom”. Skapa filen `layout.php` som inkluderar `page2.php`. `Page2.php` ska bara innehålla body-delen.



# Mer extra

- Skapa en PHP-fil som heter `functions.php`.
- I filen ska du skriva en funktion som returnerar ett framslumpat datum.  
(Se inlämningsuppgiften.)
- Skapa filen `sida.php`. Den ska innehålla en vanlig "html-grund", dvs ha en header, title, en tom body och lite sånt.
- Längst upp i `sida.php` ska du inkludera filen `funktioner.php` med `require`. Vad händer?
- Inkludera istället `funktioner.php` med `include`. Vad händer?
- Inkludera nu `functions.php` istället. I bodyn ska du skapa ett php-block som innehåller en loop som skriver ut tio framslumpade datum.

# Filhantering

- `readfile()` läser en fil.
- `echo readfile("webdictionary.txt");`

# fopen()

- fopen() ger en referens till en fil och fler alternativ.
- ```
$myfile = fopen("webdictionary.txt", "r") or die("Unable  
to open file!");  
echo fread($myfile,filesize("webdictionary.txt"));  
fclose($myfile);
```
- <http://php.net/manual/en/function.fopen.php>
- [https://www.w3schools.com/php/php\\_file\\_open.asp](https://www.w3schools.com/php/php_file_open.asp)

# fopen()

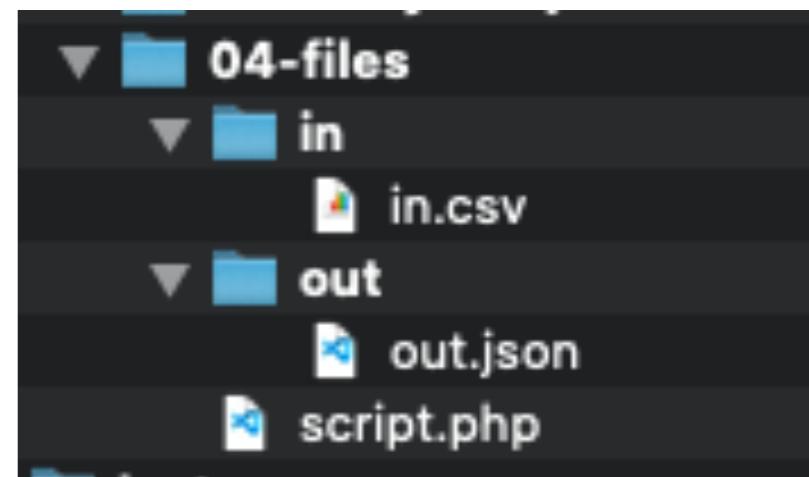
- fopen() används även för att skapa filer.
- ```
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
```
- [https://www.w3schools.com/php/php\\_file\\_create.asp](https://www.w3schools.com/php/php_file_create.asp)

# fopen()

- Man kan även öppna filer via olika protokoll direkt, t ex ftp.
- <http://php.net/manual/en/wrappers.php>

# Övning

- Skapa ett php-skript.
- Skapa en projekt-struktur som i bilden.  
[https://www.php.net/manual/en/function.fgetcsv.php](#)
- Öppna CSV-filen och läs in innehållet.  
[https://www.php.net/manual/en/function.fgetcsv.php](#)
- Gör om innehållet till JSON och spara innehållet till out.json. Skriv över den befintliga filen.  
[https://www.php.net/manual/en/function.json-encode](#)



# Ta emot formulärdata

- Skapa en html-fil send.html (eller php-fil send.php, det spelar ingen roll) som innehåller ett formulär.
- Sätt method till post och action till recieve.php.
- Se till att komponenterna har namn, t ex  
`<input type="text" name="first_name">`
- I skriptet recieve.php, skriv ut `$_POST` med `print_r()` och se vad den innehåller.
- Skriv ut en specifik komponent, t ex `$_POST["name"]`.
- Om method är satt till get kan du använda `$_GET`.

# Mer formulär

- Detta är dock inte säkert med hänsyn till exvis SQL injections och bör inte användas. Använd filter istället.
  - <https://www.php.net/manual/en/function.filter-input>
- Ex:

```
$search_html = filter_input(INPUT_POST, 'search',
    FILTER_SANITIZE_SPECIAL_CHARS);
```

**Typ av input** ↓ **Namn på komponent** ↓

↑  
**Typ av filter**

# Ladda upp filer

- ```
<form action="upload.php" method="post" enctype="multipart/form-data">
    Select image to upload:
    <input type="file" name="fileToUpload" id="fileToUpload">
    <input type="submit" value="Upload Image" name="submit">
</form>
```
- [https://www.w3schools.com/php/php\\_file\\_upload.asp](https://www.w3schools.com/php/php_file_upload.asp)

# Ladda upp filer

- ```
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}
```
- <http://php.net/manual/en/function.pathinfo.php>

# Kolla om fil finns

- // Check if file already exists  
if (file\_exists(\$target\_file)) {  
 echo "Sorry, file already exists.";  
 \$uploadOk = 0;  
}

# Begränsa filstorlek

- ```
// Check file size
if ($_FILES["fileToUpload"]["size"] > 500000) {
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
}
```

# Begränsa filtyp

- // Allow certain file formats  
if(\$imageFileType != "jpg" && \$imageFileType != "png" &&  
\$imageFileType != "jpeg"  
&& \$imageFileType != "gif" ) {  
 echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";  
 \$uploadOk = 0;  
}
- [https://www.w3schools.com/php/php\\_file\\_upload.asp](https://www.w3schools.com/php/php_file_upload.asp)

# Övning

- Skapa ett formulär som låter användaren ladda upp Word-filer och Excel-filer, men bara om de är mindre än 1MB stora.
- Döp om filerna till en timestamp, t ex 2020-01-16\_1035.docx och spara dem i mappen uploads.
- Extra: Spara filerna i en mapp med veckodagsnamnet. Ovanstående fil skulle alltså hamna i mappen thursday istället för uploads.  
<https://www.php.net/manual/en/function.date.php>
- Extra: Spara bara de senaste 5 filerna i varje mapp.  
<https://stackoverflow.com/questions/1785039/find-oldest-file-in-a-folder-using-php#answer-1785086>

# Scope

- Scope handlar om var en variabel existerar.
  - local
  - global
  - static

# Lokal

- En variabel som skapas i en funktion finns bara i den funktionen.

```
function myTest() {  
    $x = 5; // local scope  
    echo "<p>Variable x inside function is: $x</p>";  
}  
myTest();  
  
// using x outside the function will generate an error  
echo "<p>Variable x outside function is: $x</p>";
```

# global

- Används för att använda en global variabel i en funktion.

```
$x = 5;  
$y = 10;  
  
function myTest() {  
    global $x, $y;  
    $y = $x + $y;  
}  
  
myTest();  
echo $y; // outputs 15
```

- Bör användas sparsamt.

<https://riptutorial.com/php/example/2496/global-variable-best-practices>

# \$GLOBALS

- Kommer ni ihåg superglobala variabler?

- \$x = 5;  
\$y = 10;

```
function myTest() {  
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];  
}
```

```
myTest();  
echo $y; // outputs 15
```

# Static

- När en funktion är klar kommer alla dess variabler att tas bort.
- Ibland vill vi undvika det.
- ```
function myTest() {  
    static $x = 0;  
    echo $x;  
    $x++;  
}  
  
myTest();  
myTest();  
myTest();
```

# Scope

- <http://php.net/manual/en/language.variables.scope.php>
- [https://www.w3schools.com/php/php\\_variables.asp](https://www.w3schools.com/php/php_variables.asp)

# Session

- En session är ett sätt att spara information (i en variabel) som kan användas på flera sidor.
- Till skillnad mot en cookie sparar inte informationen på användarens dator.
- Istället sparas ett id som kopplas ihop med data på servern.

```
session_start();  
$_SESSION["favcolor"] = "green";  
$_SESSION["favanimal"] = "cat";
```

# Övning

- Skapa två php-skript.
  - I den ena filen sätter du ett par sessionsvariabler.
  - I den andra skriver du ut dem om de är satta.  
<https://www.php.net/manual/en/function.isset.php>
  - Skicka inte informationen genom ett formulär, kontrollera bara om sessionsvariablerna är satta och skriv ut dem. Du kan skapa en vanlig länk från den ena sidan till den andra.
- Extra: Lägg till en historik som lägger till sidan du är på som en besökshistorik. Skriv ut sidorna du har besökt som länkar. Skapa ett par sidor, inkludera "historik-funktionen", surfa runt på några sidor och testa.  
<https://stackoverflow.com/questions/6768793/get-the-full-url-in-php>

# Cookies

- `setcookie("TestCookie", $value);`
  - <https://www.php.net/manual/en/function.setcookie.php>
- I manualen skriver man ut cookies med `$_COOKIE`-variabeln. Gör inte det. Använd `filter_input` i stället.
- Cookies ingår i HTTP-headern, så `setcookie()` måste anropas innan någon data skickas till browsern.
- <https://www.guru99.com/cookies-and-sessions.html>

# Övning

- Gör ett skript som räknar hur många gånger du har läst in sidan.
- Spara talet i en cookie och skriv ut det.
- Läs in räknaren från cookien om den finns. Använd filter\_input.
- Uppdatera räknaren med ett och skriv ut det nya värdet:  
"Antalet besök: 3"
- Ladda om sidan några gånger och kolla att det fungerar.
- Extra: Lägg till fler variabler som user, ip-address och spara i cookien. Läs dokumentationen.

# Utvärdering

- Prata i grupper om 2-3 personer i två minuter.
- Vad har varit bra idag?
- Vad skulle kunna förbättras?