

Mat-IT forum

Simon Ringive
Jonas T. Kvist
Sofie Aleksandra Borup Harning
Maya Saietz

17. januar 2012

Resumé

Matematik/IT studieretningen på HTX er lidt noget for sig selv – vi er en af de studieretninger med færrest ”generelle fag” – her menes fag som fysik, kemi, samfundsfag og andre ting som man får prøvet af på grundforløbet. Mat/IT er en studieretning som man er nød til at vælge i blinde, i håb om at ”det der programmering” viser sig at være interessant. Vi er desuden den eneste klasse på skolen som har IT som fag. Det betyder at vi kun kan diskutere vores fag med vores egen klasse. Medmindre vi altså vil bruge vores sparsomme frikvarterer på at traske til den anden ende af bygningen for at snakke med 1.- og 2.g’erne. Da de fleste pauser kun varer 5-10 minutter, er der ikke rigtig tid til den slags udflugter.

Vores løsningsforslag er et online diskussionsforum hvor man kan diskutere lektier i programmering, IT og for den sags skyld også andre fag. Ud over det vil der være mulighed for generel nørdethed. For at gøre det ekstra attraktivt skal siden også have en artikelsektion hvor folk kan skrive når de har noget IT-relateret på hjerte (som de mener resten af verden gerne vil læse og have stående på siden for tid og evighed), eller hvis man bare har links til nogle interessante artikler andre steder på nettet.

Indhold

1	Projektbeskrivelse	4
1.1	E/R-diagram	4
2	Spec - MI Forum	5
2.1	Overblik	5
2.1.1	Software	5
2.2	Hypotetiske cases	6
2.3	Ikke-mål	6
2.4	Side-for-side specifikation	7
2.4.1	Forside	7
2.4.2	Forum-forsiden	8
2.4.3	Adminmenuen	9
2.4.4	Opret kategori og -forum siderne	9
2.4.5	Rediger kategori og -forum siderne	11
2.4.6	Vis kategori	11
2.4.7	Vis fora	12
2.4.8	Vis tråd	12
2.4.9	Opret tråd	13
2.4.10	Svar på tråd	14
2.4.11	Artikel-forsiden	14
2.4.12	Vis artikel	15
2.4.13	Skriv artikel	16
2.4.14	Rediger artikel	18
2.4.15	Login-siden	18
2.4.16	Registrerinssiden	19
2.4.17	Brugerliste	21
2.5	Andre scripts	21
2.5.1	util.php	22
2.5.2	config.php og install.php	22
2.5.3	auth.php	22
2.5.4	logout.php	22
2.5.5	connect.php	22
3	Afprøvning af siden	22
3.1	Typiske problemer	23
4	Konklusion	23

1 Projektbeskrivelse

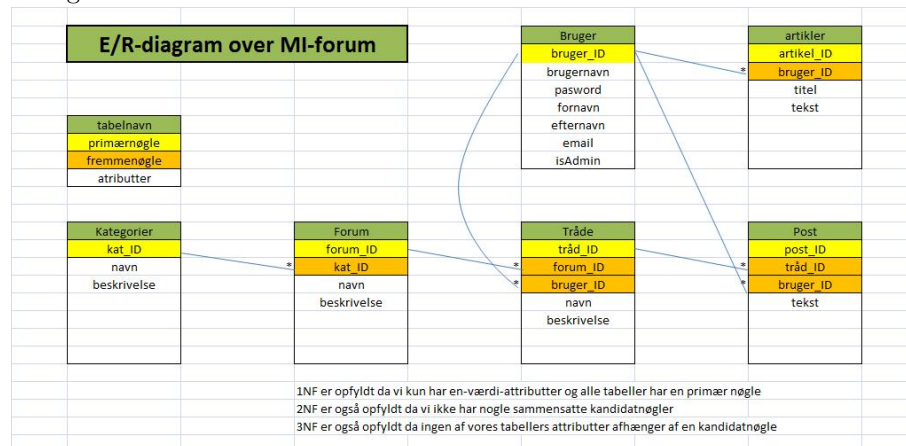
Opgaven i dette projekt var at lave en e-handel – kort fortalt et webinterface til en database. Databasen skulle være i MySQL, interfacet i xhtml, PHP og JavaScript. Ved hjælp af dette interface skulle brugere kunne bestille varer hjem, og den fiktive virksomheds ansatte skulle kunne administrere varekataloget og bestillingerne. Der var dog en ting ved denne opgavestilling vi ikke brød os om:

Fiktive virksomheder er kedelige. De findes ikke. Man kan ikke bruge dem til noget. Man kan lave så mange spændende ting med PHP og databaser som kan bruges til alt muligt nyttigt, og vi skal lave noget fiktivt? Hvorfor? Vi brainstormede og kom op med ideen at lave et online forum for vores studieretning. Det kræver de samme – hvis ikke mere komplicerede – databaser som en e-handel, og det er en noget mere interessant opgave.

Som en 4-mands gruppe blev vi nød til at dele arbejdet op (mest fordi vi fik besked på det). Simon og Jonas kom til at stå for at få forummet op og køre, Maya lavede de generelle funktioner der skal bruges på alle sider (forbind til databasen, layout-ting, menu, css) og et script til at oprette en fungerende database til siden. Sofie kom til at stå for brugerdatabase - login, autentificering, administration og den slags.

1.1 E/R-diagram

For at få et overblik over hvad vores database skulle indeholde, lavede vi et E/R-diagram.



Når man snakker om databaser findes der en række normalformer. En database kan være på første(1.), anden(2.) eller tredje(3.) normalform¹ med hensyn til relationer og opbygning af tabeller. Når man laver en database vil man helst have at den skal være på 3. normalform, da man så undgår at data gentages.

¹Ifølge Datalogi bogen af Martin Damhus

1. Normalform har to krav, for det første skal hver tabel have en primær nøgle, da man ellers ikke kan slå op i en tabel. Andet krav er at alle attributter er en-værdi attributter, man må ikke have nogle sammensatte attributter da det bare gør det sværere at få fat i dele af attributter eller ændre i dem.

Dette opfylder vores database, men der er også tale om at ingen attributter må have værdien NULL dette kaldes for stærk 1.normalform². Vi har valgt at man gerne må have Null værdier i nogle situationer fordi vi nogle steder ikke syntes det er nødvendigt at man behøver skrive f. eks en beskrivelse til en kategori.

2. Normalform har først og fremmest det krav at databasen er i 1. normalform. Derudover skal der enten ikke være nogen sammensatte kandidatnøgler (som er en nøgle bestående af to eller flere attributter), eller at der er en eller flere sammensatte kandidatnøgler, men at de attributter der afhænger af kandidatnøglen afhænger af hele den sammensatte nøgle og ikke kun noget af den. Ellers vil der kunne opstå problemer som at noget data indtastes 2 gange og skriver man forkert kan der opstå noget være rod.

Vores tabel opfylder anden normalform da vi ikke har nogen sammensatte kandidatnøgler.

3. Normalform har kun, udover selvfølgelig at være på 2. normalform, det krav, at ingen attributter indirekte afhænger af en kandidatnøgle. Det kan dog løse ved at splitte tabellen op i 2, der begge er i 3. normalform.

Dette opfylder vores tabel også. Vi har dermed opfyldt alle normalformer (med undtagelse af stærk 1. normalform).

2 Spec - MI Forum

Dette er spec'et for vores side - en grundig beskrivelse af alt hvad det skal kunne, set fra en brugers synsvinkel.

2.1 Overblik

MI Forum er et online forum for Matematik/IT studieretningen på TEC Lyngby. Det giver mulighed for at diskutere lektier, fritidsinteresser og nyheder indenfor IT. Desuden skal der være en artikel-sektion hvor folk kan skrive hvis de har noget interessant at sige.

Dette spec er ikke komplet - det vil blive redigeret undervejs, efterhånden som beslutninger bliver taget.

2.1.1 Software

Siden udvikles i xhtml, PHP, MySQL og JavaScript. Vi bruger apache-servere med MySQL og PHP, og bruger phpmyadmin til at administrere databasen.

²Ifølge Datalogi bogen af Martin Damhus.

Ud over det bruger vi et javascript-bibliotek ved navn google-code-prettify, som sørger for farvelægning af kode i kode-tags på forummet. Den eneste anden software vi bruger til udvikling er text editors - notepad++ til Windows-folkene, mens Maya bruger Vim.

Til at holde styr på vores kode og sikre at alle har den nyeste version af siden, bruger vi et version control system som hedder Git, med et online repository på www.github.com.

2.2 Hypotetiske cases

Med henblik på design af brugerinterface har vi skrevet to scenarier der kan beskrive hvordan mulige brugere kunne finde på at bruge MI Forum.

Scenarie 1: Alfred

Alfred er en 17 år gammel 2.g'er som egentlig mest valgte Mat/IT fordi han godt kan lide at spille computerspil. Han går ikke så meget op i skolen, men det lykkes ham dog at opnå middelmådige karakterer, hovedsageligt ved at deltage i grupper hvor han skriver alle de kedelige ting i rapporterne. Hans erfaring med computere begrænser sig til spil samt browsing af internettet. Han har programmeret en smule i IT, men har lidt svært ved at forstå hvad 'de der strings er for nogle'. Han bruger Windows og har aldrig prøvet andet.

Alfred kommer til MI Forum for lektiehjælp, primært i programmering og IT. Han har en tendens til bare at poste sin kode på forummet og håbe at nogen vil rette det, men efter flere gange at have erfaret at folk ikke gider lave hans lektier for ham, begynder han at poste nogle ordenlige uddybende spørgsmål. Da folk viser sig at være flinke nok, begynder han også at hænge ud i forummets off-topic sektion, hvor han bliver venner med flere elever fra de andre MI-klasser.

Scenarie 2: Natasja

Natasja er en 18 år gammel 3.g'er som elsker at programmere. Hun skriver flydende Python og har lavet websites i årevis. Hun bruger Windows, og er i stand til at finde rundt i kontrolpanelet i Windows 7. Hun er ret klog.

Natasja hænger mest ud på MI Forum for at snakke om kode. Hun har svaret på en del af Alfreds spørgsmål. Natasja skriver artikler af og til, og poster dem på MI Forum's artikel-sektion.

Efter at have været aktiv på forummet i et par måneder siger Natasja ja til et tilbud om at blive administrator. Hun får ansvar for forummets Python-sektion, men har også mulighed for at rette i andre ting.

2.3 Ikke-mål

De følgende ting er *ikke* en del af MI Forum's funktionalitet.

- Brugerprofiler
Vi laver ikke funktionalitet for at lave en stor, detaljeret brugerprofil til folk. Der er alligevel aldrig nogen der læser dem.
- Filupload
Vi laver ikke funktionalitet for at brugere kan vedhæfte filer til deres forum posts, eller på andre måder uploade filer til serveren. Vi kan overveje et bbcode tag for at indsætte billeder fra en url (altså de skal ligge på nettet i forvejen).

2.4 Side-for-side specifikation

MI Forum består af en hel del sider. Alle sider som brugere kommer i nærheden af vil følge et standard format, med et logo og en navigationsbar øverst, efterfulgt af sidens indhold. Hele siden er centreret på skærmen.

Alle sider er skrevet i en blanding af xhtml, PHP, MySQL og JavaScript.

2.4.1 Forside - index.php

Forsiden er hvad brugeren ser når han går ind på vores side. Her skal være to bokse, en for artikler og en for forummet, som viser den nyeste aktivitet. Forum-boksen skal stå øverst, da vi regner med at det er her der oftest vil være sket noget nyt. Man kan altid komme tilbage til forsiden ved at klikke på knappen 'Forside' i navigationsbaren.

Hver aktivitet beskrevet i boksene skal være et link til den pågældende forum-post eller artikel.

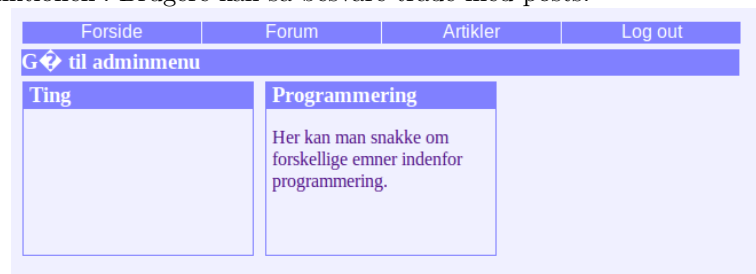


De fine bokse der viser den seneste aktivitet fungerer ved hjælp af et simpelt SQL udtræk som finder de artikler og forum-posts med højeste ID. Herefter bliver tekst-længden skåret ned og kørt igennem en smule JavaScript der sørger for at de relevante bbcode tags bliver erstattet med html. Desuden bliver der

kørt nogle regex'es for at fjerne kode-tags, da de fylder alt for meget til at have stående på forsiden.

2.4.2 Forum-forsiden - forum.php

Forummet er organiseret i kategorier, fora, tråde og posts. En kategori er et overordnet emne, fx 'Programmering' eller 'Rollespil'. Under hver kategori kan man så have fora - under Programmerings-kategorien kunne man for eksempel have fora for forskellige programmeringssprog. I et fora kan man så have tråde, som brugerne kan oprette, med emner som for eksempel 'Hjælp med strcpy() funktionen'. Brugere kan så besvare tråde med posts.



Forum-forsiden findes ved at klikke på knappen 'Forum' i navigationsbaren. Her ses en oversigt over alle de kategorier der findes i forummet. Hver kategori vises som en boks med en overskrift (titlen på kategorien) og en beskrivelse. Hver boks er et link til en side der viser den pågældende kategori.

Hvis man er logget ind som en bruger med admin-rettigheder, vises der, ovenfor kategorierne, en knap med teksten 'Gå til admin menu'. Den linker til adminmenuen, hvor man kan redigere i kategorier og fora.

Forum-forsiden bruger et simpelt SQL udtræk fra kategori-databasen, som den sorterer efter categoryID så de nyeste kategorier vises først. Den tjekker om man er logget ind som administrator ved at hente isAdmin-feltet fra databasen for den bruger hvis ID svarer til det ID der er gemt i \$_SESSION['LoggedIn']. Hvis værdien i isAdmin feltet er == 1, så er brugeren administrator (se linje 4):

```
1 $admincheck = 'SELECT isAdmin FROM users WHERE userID = "'.$_SESSION['LoggedIn'].'"';
2 $isadmin = mysql_query($admincheck) or die(mysql_error());
3 while($row = mysql_fetch_array($isadmin)){
4     if($row['isAdmin']==1){
5         // Do stuff
6     } else {
7         // Do other stuff
8     }
9 }
```

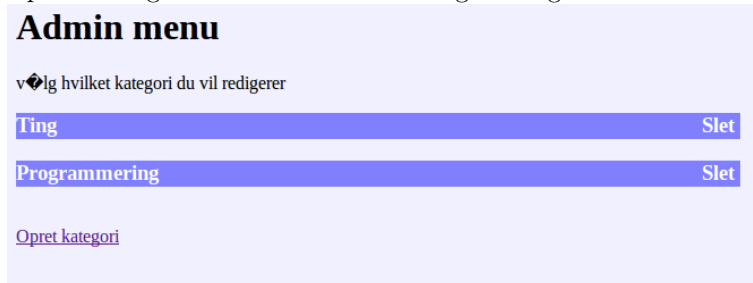
Vi skriver et @ foran sessions-variablen for at fortælle PHP at den ikke skal sende warnings hvis variablen ikke findes. Den pæne måde at gøre det på er at tjekke om variablen er sat først, men det her har samme effekt.

2.4.3 Adminmenuen - adminmenu.php

Admin menuen og dens undersider er lavet på en brugervenlig måde, så man kan fortage admin relateret ændringer (med undtagelse at gøre brugere til admin) som at oprette, redigerer og slette kategorier eller fora. På alle admin-sider tjekkes der om brugeren er logget ind som admin, før de får lov til at se siden.

Disse sider er "bag facaden" hvilket vil sige at der her ikke gøres så meget ud af bruger interfacet, men mere det funktionelle.

Selve admin-menuens forside viser en liste af alle kategorierne, og et link med teksten 'Opret kategori', som fører til 'Opret kategori'-siden. Ud for hver kategori er der en slet-knap. Klikker man på slet-knappen, popper der en lille boks op og spørger om man er sikker på at man vil slette en hel kategori. Klikker man OK bliver kategorien og alle tilhørende fora, tråde og posts slettet. Klikker man på en kategori kommer man til 'Rediger kategori'-siden.



Siden fungerer nogenlunde som forum-forsiden - i stedet for at lave pæne bokse, bliver hver kategori vist som et h3 element med en slet-knap til højre. Sletknappen er ikke et link, men kalder i stedet en JavaScript funktion som tager et ID som parameter. Funktionen laver et pop-up vindue som spørger om man er sikker på at man vil slette, og hvis man klikker OK, bliver window.location.href sat til den PHP-side der sørger for at slette kategorien.

```
1 <script type="text/javascript">
2   <!--
3       function confirmation(id) {
4           var answer = confirm("Vil du slette denne kategori?")
5           if (answer){
6               window.location.href = "sletcat.php?categoryid
7               ="+id;
8           }
9       }
10  </script>
```

2.4.4 Opret kategori og -forum siderne opretcat.php, newcat.php, opretforum.php, nyt-forum.php

Oprettelse af en ny kategori og et nyt forum foregår på samme måde, forskellen er at der er flere id'er der skal holdes styr på under oprettelse af et forum under en kategori, så man nemt kan finde den igen hvor den høre til.

På siden kan man skrive en overskrift og en beskrivelse og trykke opret kategori. Klikker man på opret kategori knappen, så oprettes kategorien, medmindre man ikke skriver noget i overskriften eller forsøger at oprette en kategori som allerede findes af samme navn. En lignende fejl meddelelse, som er beskrevet mere detaljeret under registrering af bruger længere nede, vises så øverst på siden.

Måden en kategori oprettes på er ved html attributten form action og input type= "submit" som sender den indtastede data til en behandlingsside angivet i form action. (Se linje 2)

```

1 <h1>Opret kategori</h1>
2 <form action="newcat.php" method="post">
3 <p>
4         <span class="label">overskrift:</span>
5         <input type="text" name="overskrift" size="80"/>
6         <br/>
7         <span class="label">beskrivelse:</span>
8         <textarea cols="100" rows="10" name="beskrivelse"></
          textarea>
9 </P>
10 <p>
11 <input type="Submit" value="opret kategori"/>
12 </P>
13 </form>

```

Til indtastning af overskrift bruges input type="text" og til indtastning af beskrivelse bruges textarea hvor man angiver kolonner og række (cols and rows) Når man trykker på knappen opret kategori, sendes data gemt i variablerne overskrift og beskrivelse til siden newcat.php.

```

1 <?php
2     $overskrift=mysql_real_escape_string($_POST["overskrift"]);
3     $beskrivelse=mysql_real_escape_string($_POST["beskrivelse"]);
4     $sql="INSERT INTO categories(categoryName,description)
5         VALUES('".$overskrift."','".$beskrivelse."');"
6     mysql_query($sql);
7 ?>

```

Her bliver brugerindtastningen escaped og gemt i de samme variabelnavne overskrift og beskrivelse, hvorefter de bliver indsat i tabellen ned i sql koden INSERT INTO og så værdier. Se linje 4. Kategorien er nu oprettet.

2.4.5 Rediger kategori og -forum siderne redcat.php, redigerforum.php, redcatquery.php, redforaquery.php

Disse to sider er ret så ens - rediger kategori-siden giver mulighed for at ændre en kategori's navn og beskrivelse, samt oprette, redigere og slette fora der tilhører denne kategori. Rediger forum-siden giver mulighed for at redigere navn og beskrivelse for fora.



[G ↩ tilbage](#)

Rediger kategoriens navn eller beskrivelse

overskrift:
Programmering

beskrivelse:

Fora i denne kategori

v ⚙lg hvilket forum du vil redigerer

C#

[Opret et nyt forum](#)

Disse sider bruger samme tekstredigering og næsten samme SQL som siderne for at oprette kategorier og fora. Forskellen er at vi her trækker det gamle indhold ud af databasen først, og skriver det i tekst-boksene.

2.4.6 Vis kategori - visfora.php

Når man, på forum-forsiden, klikker på en kategori, kommer man til en side der viser en oversigt over alle fora i den pågældende kategori.



Forside	Forum	Artikler	Log out
Python			
Antigravity is pretty awesome!			
C#			
Ting			

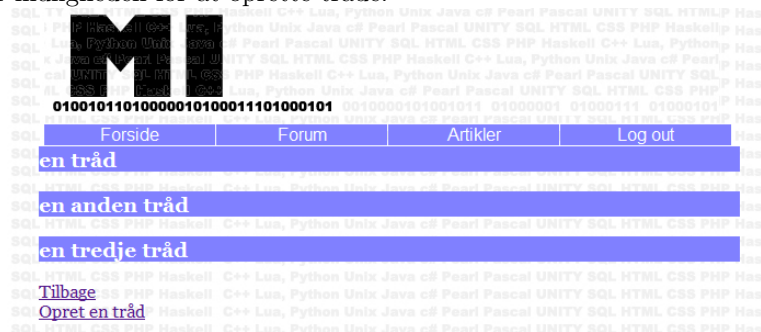
[tilbage](#)

Det er den samme PHP-fil der viser alle kategorier, den finder så de rette fora ud fra kategoriens ID, som sendes via \$_GET-array'et. Vi bruger et SQL-

udtræk til at hente alle fora hvor kategori-ID'et matcher, og viser dem med de nyeste først, på præcis samme måde som med kategorierne på forum-forsiden.

2.4.7 Vis fora - vistraade.php

Foraer bliver vist meget på samme måde som kategorier - en liste af alle tråde i det pågældende fora, sorteret med de nyeste først. Forskellen er at brugere her har muligheden for at oprette tråde.



2.4.8 Vis tråd - viewThread.php

Når man, på Vis Fora siden, klikker på en af de viste tråde, kommer man til denne side. Her vises først titlen og den tekst tråd-staren skrev, derefter alle de posts der er skrevet i tråden. Ved hver post vises navnet på den bruger der har skrevet den.

Forside	Forum	Artikler	Log out
---------	-------	----------	---------

Docstrings er for seje

Oprettet af: *Maya Saietz*

Diskussion om docstrings og deres muligheder!

```

1. def docstringExample ():
2.     """This is a docstring.
3.
4.     It spans multiple lines"""
5.
6.     # Do something
7.
8.     print(docstringExample.__doc__)

```

Sejt, huh?

Jeps, det er ret sejt!

```

1. print me.__doc__

```

Skrevet af: *Maya Saietz*

[Svar](#)
[Tilbage](#)

Nederst på siden vises der en Tilbage-knap som fører til Vis Fora siden, og, hvis man er logget ind, en Svar-knap.

Siden fungerer ved hjælp af to queries - et der finder åbningsposten i threads-tabellen, og et der finder de resterende posts i post-tabellen. Posts vises vha. et while-loop, som udskriver en bunke javascript der sørger for at formatere teksten rigtigt.

2.4.9 Opret tråd - oprettraad.php, ny-traad.php

Denne side kommer man til ved at klikke på linket 'Opret tråd' på Vis fora-siden. Her kan man skrive et nyt emne på forummet. Det foregår ved hjælp af samme JavaScript text editor som til artikel-siden.

Opret tråd

Overskrift

Beskrivelse

[b] [i] [u] [code]

opret tråd

Text editoren kan bl.a. lave `[code]` tags, som resulterer i at den tekst der står imellem vil blive stillet op med en monospace font og farvelagt alt efter hvilket sprog det er. Det gør det nemt at bruge forummet til at diskutere programmerings-problemer ved simpelthen at skrive den problematiske kode på denne måde.

2.4.10 Svar på tråd - writePost.php, createPost.php

Denne side virker præcis som Opret tråd-siden, med den undtagelse at man ikke kan skrive en titel. Den bruges til at besvare tråde, og hvad der skrives her havner i post-tabellen i vores database, hvor det kan identificeres ved trådens ID.

2.4.11 Artikel-forsiden - articles.php

Når man klikker på knappen 'Artikler' i navigations-baren kommer man til denne side. Her er alle artikler listet ved navn, og man kan klikke på dem for at læse artiklen. Nederst på siden er der et link - hvis man er logget ind står der 'Skriv ny artikel', og det fører til en side hvor man kan - hold fast - skrive en ny artikel! Hvis ikke man er logget ind, står der 'Log ind for at skrive en ny artikel', og linket fører til login-siden.

Artikler

Cookie monsters observed

This is an article

[Skriv ny artikel](#)

Siden fungerer med et SQL query der henter alle artikler fra databasen og viser dem som h3 elementer.

```
1 $get_articles = "SELECT * FROM 'articles' ORDER BY articleID DESC";
2 $result = mysql_query($get_articles) or die(mysql_error());
3
4 while($row = mysql_fetch_array($result)) {
5     echo "<h3><a href='viewArticle.php?id=".$row['articleID']."'>";
6     echo $row["title"]."<br />" ;
7     echo "</a></h3><br />";
8 }
```

2.4.12 Vis artikel - viewArticle.php

Dette er den side man bliver taget til når man klikker på en artikel på Artikel-forsiden. Her vises den pågældende artikel med titel og forfatter efterfulgt af hele teksten. Hvis man er logget ind som den bruger der har oprettet artiklen, kan man redigere eller slette den ved hjælp af to links i bunden af siden.

This is an article

af Admin Admin

It shows off some code.

```
1. print("Hello, World!");
2. function sayHello (name) {
3.     print("Hello, " + name + "!");
4. }
```

Have fun! 😊

[Rediger](#) [Slet](#)

Siden fungerer ved at kigge i databasen efter en artikel med det ID som er angivet i GET-array'et. Hvis en sådan ikke findes, bliver man sendt tilbage til artikel-forsiden. Ellers bliver artiklen vist. Desuden sammenlignes artiklens writerID med den nuværende brugers ID, og hvis de er ens (eller hvis brugeren er administrator) vises links for at redigere og slette artiklen. Her er et par af de mere interessante ting fra koden:

```
1 $getAuthor = "SELECT firstname, lastname FROM users WHERE userID = '".
    $row['writerID']."'";
2 $writerResult = mysql_query($getAuthor) or die(mysql_error());
```

```

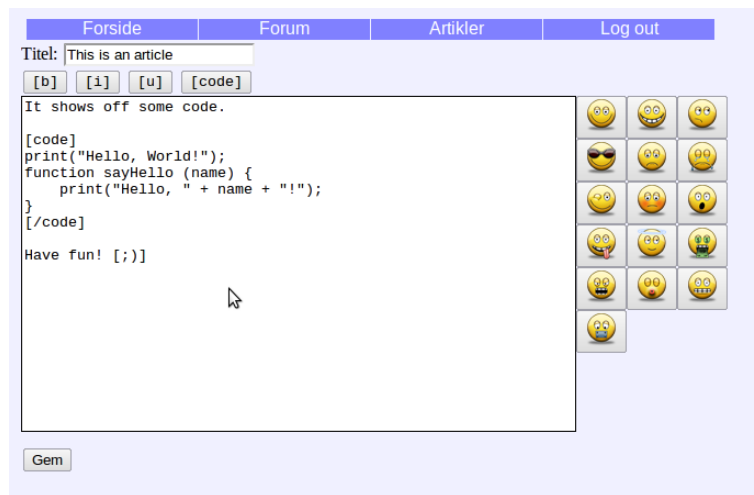
3
4 // Get name of author
5 while($writer = mysql_fetch_array($writerResult)) {
6     $authorName = $writer['firstname'] . ' ' . $writer['lastname'];
7 }
8
9 // Pretty printing of the article
10 echo '<h2>'. $row['title']. '</h2>';
11 echo '<i>af '. $authorName. '</i><br /><br />';
12 echo '<script type="text/javascript" src="google-code-prettify/src/
    prettify.js"></script>';
13 echo '<script type="text/javascript" src="text_edit.js"></script>';
14 echo '<script type="text/javascript">';
15 echo 'window.onload = function () {prettyPrint(); }';
16 echo 'var textToWrite = "'.mysql_real_escape_string($row['text']).'";';
17 echo 'textToWrite = view_text(textToWrite);';
18 echo 'document.write(textToWrite);';
19 echo '</script>';
20
21 // Check if user can edit and delete
22 if(isset($_SESSION['LoggedIn'])) {
23     $isAdmQuery = "SELECT isAdmin FROM users WHERE userID = ".
        $_SESSION['LoggedIn']. ";";
24     $isAdm = mysql_query($isAdmQuery) or die(mysql_error());
25     $admRow = mysql_fetch_array($isAdm);
26     if($_SESSION['LoggedIn'] == $row['writerID'] || $admRow['
        isAdmin'] == 1) {
27         echo "<br /><br /><span style='float: right;*><a href='
            editArticle.php?id=".$row['articleID']."'>Rediger</
            a>
28         <a href='deleteArticle.php?id=".$row['articleID']."'>
            Slet</a></span><br />";
29     }
30 }

```

Dette script involverer, som kan ses i ovenstående kode-listing, et grimt hack hvor PHP udskriver JavaScript. Det er nødvendigt for at kunne behandle vores hjemmelavede bbcodes og formatere koden pænt.

2.4.13 Skriv artikel - writeArticle.php, createArticle.php

Denne side giver mulighed for at skrive nye artikler til siden. De bliver så vist under artikel-sektionens forside, og på forsiden når de er nye. Den fungerer ved hjælp af en text-editor skrevet i JavaScript. Dette giver mulighed for at bruge nogle hjemmelavede bbcodes, enten ved at indtaste dem eller ved at trykke på knapperne som findes ovenfor og til højre for tekstboksen.



Selve det JavaScript det sørger for at text-editoren virker er for langt til at inkludere her, men kan ses i text_edit.js. Et uddrag der dog er interessant at kigge på er funktionen for at sætte tags rundt om et markeret stykke tekst:

```

1 function mod_selection (val1,val2) {
2     // Get the text area
3     var textArea = document.getElementById('my_text');
4
5     // IE specific code.
6     if( -1 != navigator.userAgent.indexOf ("MSIE") ) {
7
8         var range = document.selection.createRange();
9         var stored_range = range.duplicate();
10
11         if(stored_range.length > 0) {
12             stored_range.moveToElementText(textArea);
13             stored_range.setEndPoint('EndToEnd', range);
14             textArea.selectionStart = stored_range.text.
15                 length - range.text.length;
16             textArea.selectionEnd = textArea.selectionStart
17                 + range.text.length;
18         }
19     }
20     // Do we even have a selection?
21     if (typeof(textArea.selectionStart) != "undefined") {
22         // Split the text in three pieces - the selection, and
23         // what comes before and after.
24         var begin = textArea.value.substr(0, textArea.
25             selectionStart);
26         var selection = textArea.value.substr(textArea.
27             selectionStart, textArea.selectionEnd - textArea.
28             selectionStart);
29         var end = textArea.value.substr(textArea.selectionEnd);
30
31         // Insert the tags between the three pieces of text.
32         textArea.value = begin + val1 + selection + val2 + end;
33     }
34 }

```

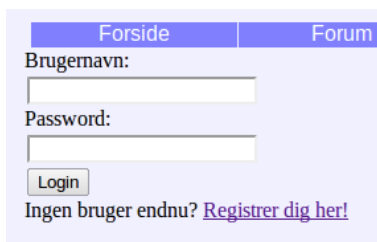
Den deler teksten i tre dele, sætter de angivne tags ind og sætter det hele sammen til en lang string igen. Bemærk den IE-specifikke kode - IE har ikke nogen funktion for at finde start og slut på en markering, så man er nød til at beregne placeringen ud fra længden af den markerede tekst, og så sætte de to variable manuelt. Derefter kan man fortsætte som normalt.

2.4.14 Rediger artikel - `editArticle.php`, `deleteArticle.php`

Denne side fungerer mere eller mindre som Skriv artikel, med den undtagelse af at den henter den gamle tekst og titel fra databasen først, og at den bruger UPDATE frem for INSERT INTO. Desuden er den kun tilgængelig hvis man er logget ind som den bruger der har skrevet artiklen eller en admin.

2.4.15 Login-siden - `login.php`, `loginBackend.php`

Dette er stedet hvor man logger ind som bruger. Man kan komme herhen ved at klikke på knappen 'Log ind' i navigationsbaren, eller via et af de mange links som findes på steder hvor vi regner med at det er nyttigt at logge ind, så som artikel-forsiden hvor man kan logge ind for at skrive en ny artikel.



The screenshot shows a login form with a light blue background. At the top, there are two navigation links: 'Forside' and 'Forum'. Below these, the form asks for 'Brugernavn:' (Username) and 'Password:'. Each label is followed by a text input field. Below the password field is a 'Login' button. At the bottom of the form, there is a link that says 'Ingen bruger endnu? [Registrer dig her!](#)'.

Siden består af en standard login form, hvor brugeren kan indtaste sit navn og løsen, og en knap med teksten 'Login'. Når man trykker på knappen, bliver indtastningerne tjekket i forhold til databasen, og hvis de passer, bliver man logget ind. Hvis man ikke har en bruger er der et link til registreringssiden lige under login-formen.

Hvis man indtaster en kombination af brugernavn og adgangskode som ikke findes i databasen, vises fejlmeddelelsen 'Forkert brugernavn eller adgangskode' i rød tekst. Hvis man mangler at udfylde et af felterne, vises fejlmeddelelsen 'Manglende brugernavn eller adgangskode' i rød tekst.

Registrer dig her!'." data-bbox="379 155 609 285"/>

Login-formen er en ganske almindelig html form, hvis submit-knap sender os til et backend-script som så tager sig af SQL-koden. Til at tjekke om brugere findes, bruger vi følgende SQL query:

```
1 $checkLogin = "SELECT * FROM users WHERE user='".  
    mysql_real_escape_string($_POST['username'])."' AND password='".md5  
    (mysql_real_escape_string($_POST['password']))."'";
```

Bemærk at vi kører md5 *efter* at have kørt mysql_real_escape_string() - rækkefølgen skal være den samme som på registreringssiden. Egentlig er det ikke nødvendigt at escape løsenet, da det ikke kan gøre nogen skade i krypteret form, men det er en god vane aldrig at putte ting i queries uden at have kørt dem igennem en escape-funktion.

Hvis man logges ind succesfuldt, redirectes man til index-siden. Ellers bliver man sendt tilbage til login-siden med en fejlkode - denne metode vil blive beskrevet nærmere under afsnittet om registreringssiden.

2.4.16 Registreringssiden - register.php, registerBackend.php

Her kan nye brugere registrere sig. Dette foregår ved at udfylde en række felter med ting som brugernavn, løsen, email, navn m.v. Der er desuden en knap med teksten 'Registrer', hvor brugeren kan klikke når hun har udfyldt alle felter.

Forside	Forum
Desired username:	
<input type="text" value="Tayacan"/>	
Password:	
<input type="password" value="*****"/>	
Retype password:	
<input type="password" value="*****"/>	
First name:	
<input type="text" value="Maya"/>	
Last name:	
<input type="text" value="Saietz"/>	
E-mail:	
<input type="text" value="mayasaietz@gmail.com"/>	
<input type="submit" value="Submit"/>	

Hvis der er felter som ikke er udfyldt, vises fejlmeddelelsen 'Venligst udfyld alle felter'.

Siden består af en html form som håndterer brugerinput og en stump PHP-kode som håndterer fejlmeddelelser. Vi benytter os af den form for fejlhåndtering mange steder i koden, så her er et eksempel:

```

1 if(!empty($_GET['error'])) {
2     if($_GET['error'] == 1){
3         echo "<span style='color:red;'>Please fill out all
4             fields.</span><br /><br />";
5     } else if($_GET['error'] == 2) {
6         echo "<span style='color:red;'>Password doesn't match -
7             did you mistype?</span><br /><br />";
8     } else if ($_GET['error'] == 3) {
9         echo "<span style='color:red;'>Username already exists
10            :(</span><br /><br />";
11     }
12 }

```

Som man kan se på koden bruger vi GET-array'et til at sende vores fejlmeddelelser.

Submit-knappen på registreringssiden leder til vores backend script, som sørger for at tjekke at input er gyldigt, krypterer koden, og kører de relevante SQL queries. Desuden sørger det for at sanitere alt brugerinput med `mysql_real_escape_string()`.

Hvis input nu *ikke* er gyldigt, sender scriptet os tilbage til registreringssiden med en fejlkode, sådan her:

```

1 if($_POST['password1'] != $_POST['password2']) {
2     header('Location: register.php?error=2');
3     exit;
4 }

```

2.4.17 Brugerliste

Her kan man se brugere, og hvis man er admin kan man gøre andre til admins.

Brugernavn:	Fornavn:	Efternavn:	Email:	Admin:
admin	Admin	Admin	admin	ja
Tayacan	Maya	Saietz	mayasaietz@gmail.com	nej

Brugernavn:	Fornavn:	Efternavn:	Email:	Admin:	Gør til admin
admin	Admin	Admin	admin	ja	
Tayacan	Maya	Saietz	mayasaietz@gmail.com	nej	<input type="button" value="Make Admin"/>

Øverst vises tabellen for en almindelig bruger, eller hvis man ikke er logget ind, nederst ses tabellen for admin-brugere.

Brugerne bliver vist ved at hive fat i dem med et MySQL-query der også ordner dem alfabetisk efter brugernavn. Herefter lægger vi dem ind i en tabel ved hjælp af et while-loop.

For at få vist 'gør admin'-knappen bruger jeg følgende kode:

```
1 <?php if($isAdmin == 1){
2     ?>
3     <td>
4     <?php
5     if($row['isAdmin'] != 1) {
6     ?>
7         <form action="users.php" method="post">
8             <input type="hidden" name="userId"
9                 value="<?php echo $row['userID']; ?>" />
10            <input type="submit" value="Gør til admin" />
11        </form>
12    <? } ?>
13    </td>
14    <?php
15 }?>
```

Først tjekker vi om vedkommende der ser siden er admin eller ej, hvis vedkommende ikke er, så bliver resten af koden bare ignoreret. Hvis vedkommende derimod er admin, tjekkes der om den bruger man er ved at placere i tabellen er admin. Hvis brugeren er det, bliver der bare lavet et tomt felt, da man jo ikke skal gøre en admin admin igen. Hvis brugeren derimod ikke er admin, oprettes en knap der gemmer brugerens userID. Således er hver knap linket til den retmæssige bruger, og når man klikker på knappen, gøres vedkommende til admin.

2.5 Andre scripts

De følgende scripts har været af betydning for hele siden, men er ikke noget brugeren lægger mærke til.

2.5.1 util.php

Dette script indeholder en række funktioner som frit kan bruges overalt på siden. For eksempel bruges funktionerne `top()` og `bottom()` på alle sider, til at lave menuen, headeren, og `bottom()` til at afslutte tags. Alt synligt indhold skal placeres mellem kald til disse to funktioner.

2.5.2 config.php og install.php

Disse to scripts tager sig af at oprette og konfigurere databasen. Man åbner `install.php` i sin browser, indtaster sit brugernavn og password til MySQL, og så opretter `config.php` automatisk en database samt en bruger som har fuld rettighed til at ændre i den, men kun den, database. På den måde sikres det at selv hvis siden skulle blive hacket, går det kun ud over den ene database.

2.5.3 auth.php

Dette script skal `require's` på alle sider som kræver at man er logget ind. Hvis en bruger som ikke er logget ind navigerer til en side hvor `auth.php` er med, vil browseren automatisk blive redirectet til login-formen.

2.5.4 logout.php

Dette er et script som log ud-knappen linker til. Det sørger for at slette alle sessionsvariable, og forsøger derefter at sende en til den side hvor man var da man loggede ud.

2.5.5 connect.php

Dette script står for at forbinde til databasen. Alle sider som skal have med databasen at gøre, behøver kun at include dette script for at have forbindelse.

3 Afprøvning af siden

Vi har under arbejdet med siden afprøvet den konstant. Jo mere man laver mellem hver afprøvning, jo sværere er fejlene at finde, så hvis vi kan lave en lille ting og sikre os at det virker, og så bruge den lille ting til at bygge noget større, er det bedst. For eksempel starter man måske med at sikre at man får fat i de rigtige værdier fra sin database ved bare at printe dem ud, inden man begynder at vise dem pænt - på den måde ved man at det er visningen der er gået galt hvis der pludselig ikke dukker noget op på skærmen.

Vi har eksperimenteret med alle former for manglende og korrump input vi kunne komme i tanke om, bl.a. html tags, PHP kode og SQL injections, og mener efterhånden at have sikret vores side mod alle angreb fra den front. Ud over det bruger vi `$_POST` frem for `$_REQUEST` når vi skal bruge variable fra POST-array'et, så brugere ikke kan taste interessante ting i URL'en.

3.1 Typiske problemer

Dette er listen over de problemer vi blev ved med at løbe ind i gennem arbejdet med siden. De kunne være nyttige at huske på til fremtidige projekter.

- At skrive POST med småt.
\$_post er ikke det samme.
- At glemme underscore foran post.
\$POST virker heller ikke.
- At glemme at sende værdier med GET-array'et.
Vi har haft masser af 'undefined index'-fejl.
- At glemme at køre sit SQL query.
- At glemme at rette variabelnavne i copy-pastet kode.
"Jamen det virkede på den anden side!"

4 Konklusion

Alt i alt synes vi at vores projekt gået rigtig godt. Som gruppe har vi disponeret vores tid og delt arbejdsopgaver ud, samt løst problemerne der nu måtte dukke op undervejs. Vi har fra idé skabt et fungerende forum, og opfyldt alle kravene fra projektbeskrivelsen. Alt dette med det resultat, at hvis vi havde midlerne, kunne forummet ligges op online, klar til brug.