Informationsteknologi eksamensprojekt: HCØL-klubber

Rapport af	Klasse
Thomas Hoffmann Kilbak	**3B1**
Skole	Fag
H. C. Ørsted Gymnasiet	**Informationsteknologi B**
Lærer	Afleveringsdato
Anders Juul Refslund Petersen	**12/04/2019**

Indhold

Indledning 2

Problemstilling 3

Projektafgrænsning 3

Formål og krav 3

Tidsplan 4

Arbejdsproces 4

Værktøjer 4

- Sprog 4
- Text-editor 5
- Xampp 6
- phpMyAdmin 6

Redegørelse for funktionalitet 7

- database.php 7
- index.php 8
- klubOversigt.php 8
- tilmeld.php 8
- frameld.php 8
- lavKlubSide.php 9
- lavKlub.php 9
- styles.css 9

Tabeller 9

ER-diagram 11

Layout og design 12

Test 12

Sikkerhed 15

Eventuelle forbedringer 16

Tilføje if-statement 16

Spam-beskyttelse 16

Bedre opbevaring af session-værdier 17

Ekstra features 17

Vurdering og konklusion 18

Bilag (kode) 19

database.php 19

index.php 20

klubOversigt.php 21

tilmeld.php 22

frameld.php 22

lavKlubSide.php 23

lavKlub.php 23

styles.css 24

Indledning

Dette er eksamensprojektet i informationsteknologi af Thomas Hoffmann Kilbak. Projektet tager udgangspunkt i, at der skal laves et IT-produkt, som skal gøre det nemmere for elever at melde sig til skolens klubber. Derudover vil der blive redegjort for den IT som er blevet benyttet til at udarbejde produktet.

Produktet blev udviklet i samarbejde med Phillip Skovbon Mikkelsen (3B1). Rapporten er dog lavet af kun mig, med undtagelse af diagrammer, tidsplanen og skærmbilleder, som vi begge benytter i hver vores rapport.

Problemstilling

På H. C. Ørsted Gymnasiet i Lyngby er der mange klubber som elever kan melde sig ind i. Disse klubber skaber sammenhold på skolen, og er gode for elevernes sociale samvær. Der er dog ikke noget sted hvor man kan danne sig et nemt overblik over hvilke klubber der er. Dette hæmmer også udviklingen af nye klubber. Mange elever vil nok ikke gå gennem besværet at reklamere for en klub med plakater på skolen, så hvis dette kunne gøres nemt, ville flere sociale fællesskaber nok opstå.

Skolens benyttede IT-service er Lectio. Denne har ikke nogen funktion til at administrere klubber, så en ekstern løsning er nødvendig. Dette er ikke kun et problem på HCØL, men reelt et problem som mange skoler, uddannelsesinstitutioner, og foreninger har.

Projektafgrænsning

Der skal laves et IT-produkt hvor man nemt kan se hvilke klubber der er, oprette nye klubber, samt at tilmelde sig klubber.

Formål og krav

Det er formålet med produktet at det skal være brugervenligt og simpelt, så flest muligt elever vil engagere sig i det sociale samvær på skolen. For at produktet skal kunne bruges, skal det dog opfylde nogle specifikke krav.

Man skal kunne…

- Logge ind med sin unikke konto.
- Oprette sig som ny bruger.
- Danne sig et overblik over hvilke klubber der er.
- Læse hvad de enkelte klubber går ud på.
- Tilmelde sig en klub.
- Framelde sig en klub.
- Oprette en ny klub.

Hvis disse krav bliver opfyldt, vil produktet være en succes. Man kan tilføje yderligere features til produktet for at videreudvikle det. Dette bliver diskuteret sidst i rapporten.

Tidsplan

Der var i alt 30 undervisningstimer afsat til dette projekt. Derudover skulle der dog også bruges tid udenfor skolen, for at kunne færdiggøre projektet i tide. Der blev derfor lagt en plan for hvornår enkelte dele skulle være nået. Planen tager udgangspunkt i de 30 undervisningstimer. Hvis en del af produktet ikke blev nået i skoletiden, blev der brugt tid uden for skolen for at indhente tidsplanen. Tidsplanen kan ses på figur 1.

Timer	**Skal færdiggøres**
0-5	Gruppedannelse, idegenerering, udvælge ide, lægge tidsplan.
5-10	Oprette database og tabeller i MySQL.
10-15	Login-system, 'opret klub'-funktion.
15-20	Klubtabel på hovedsiden (klubOversigt.php)
20-25	Tilmeld/frameld funktionalitet, design af brugerflade (CSS).
25-30	Endelige tests, rapportskrivning.

Figur 1: Tidsplan for projektet.

Arbejdsproces

Arbejdet med produktet blev generelt uddelegeret mellem gruppens to medlemmer. Vi arbejdede for det meste i samme rum, hvilket gjorde at det ikke var nødvendigt at lægge filerne online. Hvis vi havde arbejdet hver for sig eller hvis vi havde arbejdet flere personer sammen, kunne det have hjulpet på arbejdet at have et GitHub repository. Dette fungerer som lagring i skyen, så alle altid kan tilgå koden og når der ændres i den, ændres det for alle.

Værktøjer

Til projektet er der gjort brug af nogle værktøjer. Disse var enten essentielle for produktets tilbliven, eller også gjorde de arbejdet nemmere og mere effektivt.

Sprog

Der er blevet brugt fire forskellige programmeringssprog: HTML, CSS, PHP og SQL. Mest prominent af disse er PHP. Jeg benytter dette sprog, da jeg har benyttet det i andre projekter, og da det er

nemt at benytte med HTML. Man kan nemlig integrere HTML i en PHP-fil ved brug af 'echo'. Dette bringer mig videre til HTML. Dette danner grundlag for den brugerflade som brugeren vil interagere med. HTML er benyttet til bl.a. at lave tabeller, knapper og tekstfelter. Udseendet af hjemmesiden er så modificeret med CSS. Dette er et sprog som gør det muligt at bestemme hvordan forskellige elementer. Det har altså ikke noget med produktets grundfunktionalitet at gøre, det er i stedet primært æstetisk.

SQL blev benyttet til at håndtere databasen. SQL gør det muligt at interagere med en database. Hvor CSS og HTML er client-side er SQL altså i stedet server-side. SQL bliver brugt når der skal tilføjes ellers fjernes noget fra databasen.

Text-editor

Der er for det meste benyttet Sublime Text som text-editor. Dette er fordi jeg synes at det er den bedst egnede til at skrive PHP, da den har gode farvekoder til forskellige dele af koden. I starten af projektet blev der benyttet Brackets, men dette program er ikke så egnet til PHP, da det tilegner langt de fleste dele af koden den samme farve.

Herunder på figur 2 ses hvordan tilmeld-funktionen ser ud i Brackets. Det kan ses, at meget af koden er orange, hvilket gør det svært at danne sig et overblik.

```
function tilmeld($connection, $userid, $clubid){
    $sql ="INSERT INTO clubDB.userclub (`user_id`, `club_id`) VALUES (".$userid.",
    ".$clubid.");";

    $tbCreated = $connection->query($sql);

    }
```

Figur 2: Hvordan tilmeld-funktionen ser ud i Brackets.

På figur 3 ses hvordan denne funktion ser ud i Sublime Text. Det kan ses, at forskellige dele af koden er fremhævet med forskellige farver, hvilket gør det nemmere at danne sig et overblik.

```
function tilmeld($connection, $userid, $clubid){
    $sql ="INSERT INTO clubDB.userclub (`user_id`, `club_id`) VALUES (".$userid.",
        ".$clubid.");";

    $tbCreated = $connection->query($sql);

    }
```

Figur 3: Hvordan tilmeld-funktionen ser ud i Sublime Text.

At benytte Sublime Text, gjorde det væsentligt nemmere at udarbejde projektet, da det var nemmere at finde fejl.

Xampp

Xampp er et program, som gør det muligt at starte en server lokalt på computeren. Dette er nødvendigt for at teste php. Man kan sætte sit eget kodeord, og når man har startet Apache- og MySQL-delene, kan man teste sin php-kode som om det befandt sig på en hjemmeside ved at gå ind på *localhost/index.php* hvis man altså ønskede at tilgå index.php.

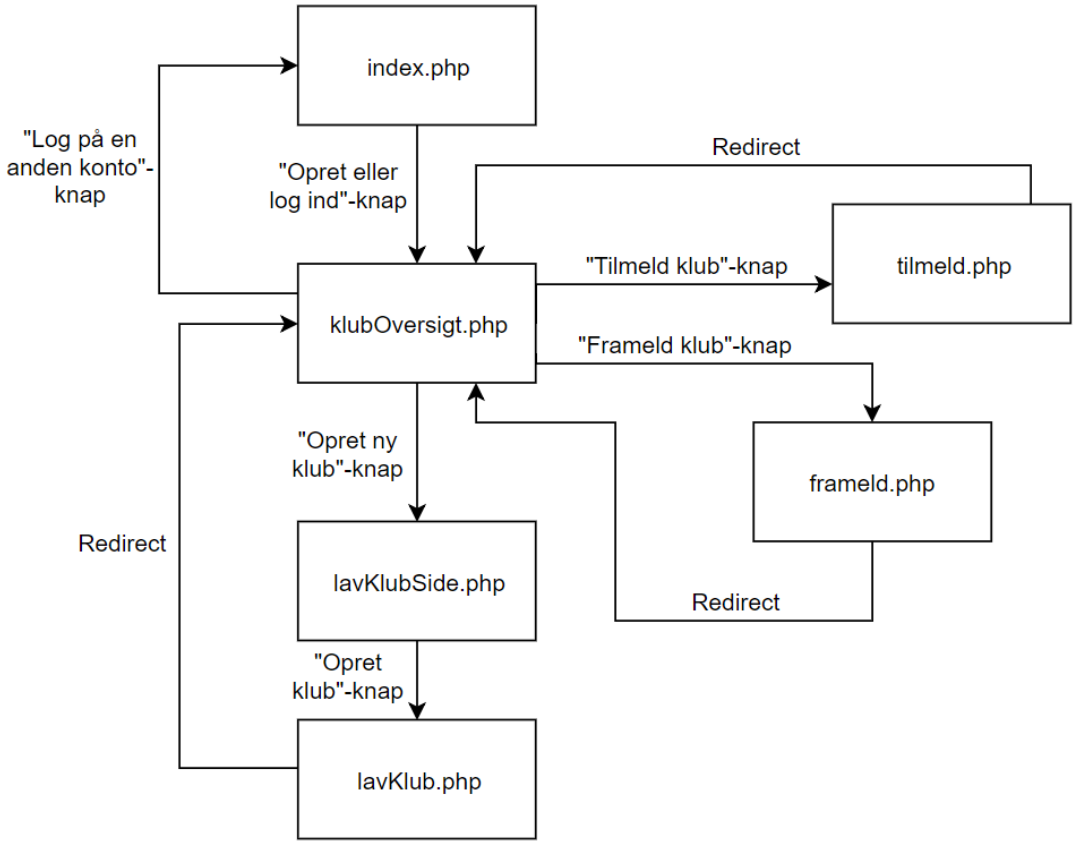
Uden xampp eller et lignende program, ville det være meget sværere at teste php, da man ellers skulle sætte en online server op.

phpMyAdmin

Dette er et browser-baseret kontrolpanel, hvor man kan administrere databaser. Dette redskab var nødvendigt, for at vi kunne tjekke om vores SQL-database var blevet oprettet korrekt. Vi kunne også bruge kontrolpanelet til at redigere i tabellerne, hvis der var sket en fejlindtastning.

Redegørelse for funktionalitet

Produktet består af en række filer. Jeg vil her redegøre for hvad hver enkelt fils rolle er. For dybere beskrivelse af koden, kan koden findes som bilag hvor der er skrevet kommentarer til de forskellige elementer. Først ses på et flowchart over hvordan man navigerer på siden. Dette kan ses på figur 4. Man starter ved login-siden index.php. Så klikker man på ”Opret eller log ind”-knappen, hvorved man bliver sendt videre til klubOversigt.php. Resten af flowchartet følger dette design.



Figur 4: Flowchart over hvordan man navigerer på siden.

Nu følger en beskrivelse af hvad hver fil gør.

database.php

Denne fil bliver ikke vist visuelt. I stedet bliver den kaldt i mange af de andre filer i form af *include ”database.php”*. Dette er fordi den indeholder funktionerne som interagerer med SQL, bl.a. opretBruger() og tilmeld(). Derudover opretter den forbindelse til databasen, og opretter selve

databasen og dens tabeller hvis de ikke findes. At have en fil som indeholder alle disse funktionaliteter, gør at andre filer kan have pænere orden, da man blot kan kalde funktionen.

index.php

Dette er den første side man møder, hvilket er grunden til at den tildeles navnet index. På denne side skal brugeren logge ind. Denne side gør kun brug af HTML (og CSS). Ved hjælp af en form kan den poste de inputværdier man giver ind i klubOversigt.php.

klubOversigt.php

Dette er den primære side af produktet. Jeg vil i denne rapport ofte referere til denne side som ”hovedsiden”. Siden starter med at bedømme om der er tale om en ny bruger eller en eksisterende bruger. Dette gøres ved at kalde funktionerne rigtigKode(), eksisterendeBruger() og opretBruger() fra database.php. Dernæst opstilles dataene fra club-tabellen. Dette gøres ved et while-loop. Den kører dermed igennem arrayet og udskriver hver gang værdierne for den givne klub. Derudover sættes der ”Tilmeld” og ”Frameld”-knapper ind.

tilmeld.php

Når man klikker på tilmeld-knappen ud for en klub, navigeres man til en side som er givet på den måde som det ses på figur 5.

```
tilmeld.php?club_id=".$row['id']
```

Figur 5: Den URL man navigeres til når man klikker "tilmeld". Bemærk at club_id bestemmes ud fra hvilken række klubben var på.

Dette gør at man bliver navigeret til tilmeld.php, og club_id sættes ind. Dette gøres ved at kalde tilmeld-funktionen fra database.php. Denne gør at user_id og club_id indsættes i userclub-tabellen, hvorved der skabes en forbindelse mellem en bruger og en klub. Dermed er brugeren tilmeldt. Tilmeld-funktionen blev vist tidligere i afsnittet ”Text-editor”.

frameld.php

Denne fungerer meget ligesom tilmeld.php, dog med den undtagelse at der kaldes frameld-funktionen fra database.php i stedet. Denne funktion finder hvor det givne club_id og det nuværende user_id er kombineret i userclub-tabellen, og sletter den entry. Dermed er man meldt ud af en klub.

lavKlubSide.php

Man navigeres hen til denne side når man vil oprette en klub. Kodemæssigt fungerer den meget ligesom index.php, da den kun indeholder HTML, og den benytter en form hvor man poster værdierne videre til lavKlub.php.

lavKlub.php

Denne fil er ikke visuel, i stedet tilføjer den kun funktionalitet. Den modtager værdierne fra lavKlubSide.php og indsætter dem i club-tabellen. Derefter dirigerer den en tilbage til klubOversigt.php.

styles.css

Dette fungerer som et stylesheet. Det vil sige at al CSS er skrevet i denne fil, og så kaldes den i de andre filer. Stylesheetet bliver kaldt i headeren på siderne. Hvordan kan ses på figur 6.

```
<link rel="stylesheet" type="text/css" href="styles.css">
```

Figur 6: Hvordan et stylesheet kaldes. Denne linje skal placeres i <head>.

Med dette stylesheet, bliver sidens design lavet. Det er denne fil, som definerer bl.a. hvordan skrifttyper og knapper skal se ud. Alle visuelle sider er sat ind i . Denne class er den som bl.a. gør at siden placeres midt på vinduet.

Ved at bruge et stylesheet, gør man de andre filer mere ordentlige, da der ikke skal bruges plads på CSS. Det gør også at man ikke gentager kode.

Tabeller

I databasen er der tre tabeller. Den første tabel er 'users'-tabellen. Denne holder brugernes navne og deres kodeord. Derudover er hver bruger tilegnet et id, så man kan referere til dem i andre tabeller. På figur 7 herunder, ses hvordan users-tabellen ser ud i phpMyAdmin.

id	navn	password
1	Mathias Poulsen	1234
2	Jens Hansen	hahaha
3	Tobias Nielsen	skole123

Figur 7: Users-tabellen som den ser ud i phpMyAdmin.

Den anden tabel hedder 'club'. Denne holder alle klubberne som er tilgængelige. Klubberne tilegnes værdier som er lignende de værdier som brugerne har. Hver klub tilegnes et navn samt en beskrivelse. Derudover tilegnes de også hver et id, så man nemmere kan referere til dem i andre tabeller. På figur 8 herunder ses hvordan tabellen ser ud i phpMyAdmin.

id	navn	beskrivelse
1	Skakklub	Skak efter skole om torsdagen
2	Rollespilsklub	Rollespil en gang om måneden
3	Astronomiklub	Kigger på planeter og stjerner

Figur 8: Club-tabellen som den ser ud i phpMyAdmin.

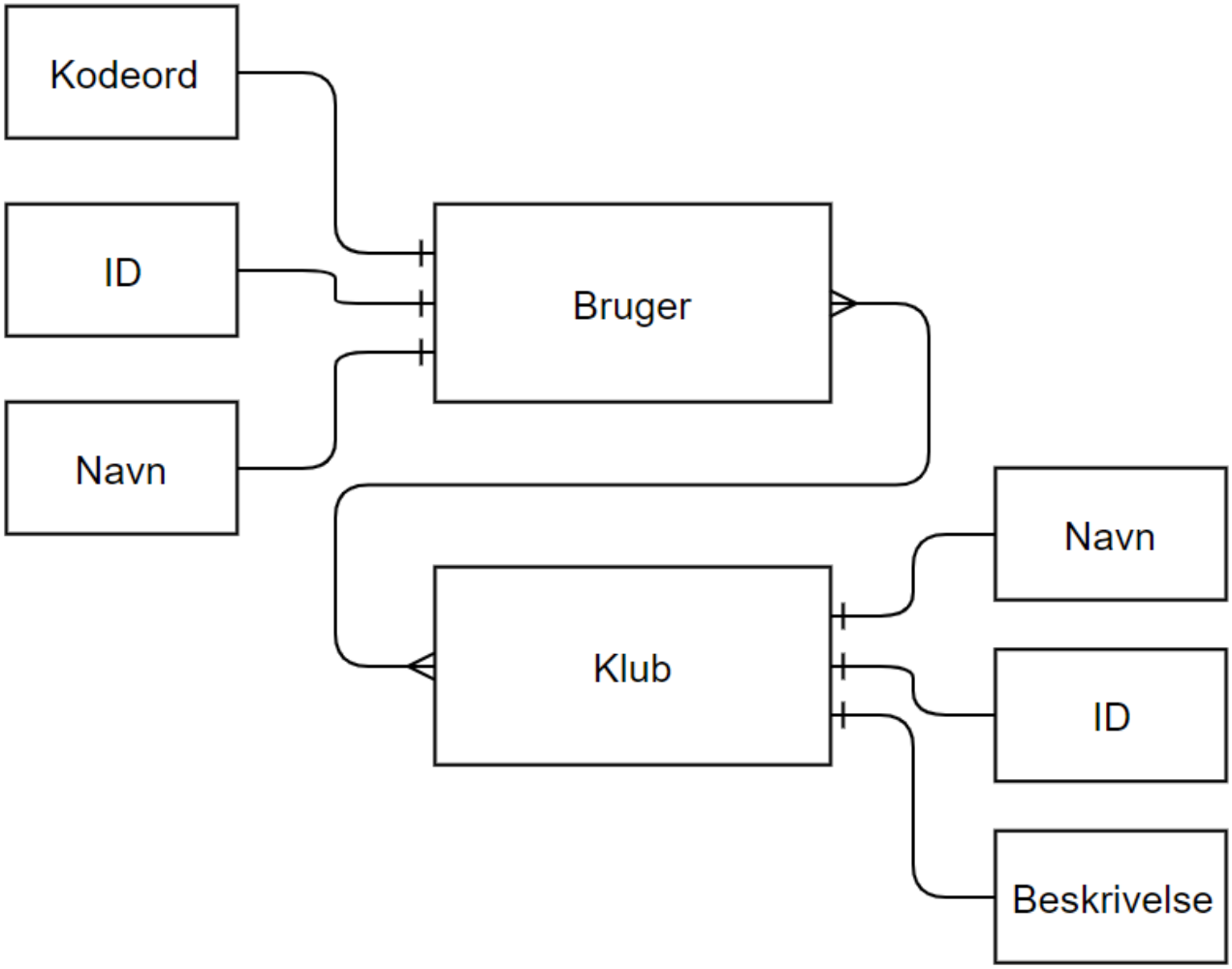
Den tredje tabel hedder 'userclub'. Denne står for at skabe en forbindelse mellem brugere og klubber. Når en bruger melder sig ind i en klub, indsættes brugerens user_id og klubbens club_id i tabellen. Det vides dermed at den bruger er med i den klub. Hvordan tabellen ser ud, kan ses på figur 9. Det kan ses, at en bruger kan være med i flere klubber, og at en klub kan have flere medlemmer.

user_id	club_id
3	1
3	6
3	5
3	2
1	1

Figur 9: userclub-tabellen som den ser ud i phpMyAdmin

ER-diagram

Programmets database gør brug af mange-til-mange-relationer. Dette giver mening, da en klub kan have mange brugere, og en bruger også kan være medlem af mange klubber. Dette kan visualiseres med et ER-diagram. ER står for ”entity relationship”. Det betyder altså hvordan forholdet mellem forskellige dele af databasen er. ER-diagrammet for produktet kan ses på figur 10. Ud fra stregerne kan det ses, at hver bruger har et kodeord, ID og navn. Hver klub har et navn, ID og beskrivelse. Ud fra stregerne kan det ses at der er et mange-til-mange forhold mellem brugere og klubber.



Figur 10: Entity relation-diagram for projektet.

Layout og design

Da produktet har et meget simpelt formål, er det vigtigt at funktionaliteten er meget nem at tilgå. Layoutet af hovedsiden (klubOversigt.php) har derfor fokus på at det skal være nemt at melde sig ind i klubber samt at oprette nye klubber. Som tidligere nævnt, benyttes CSS til at styre sidens layout. Alting på siden centreres, så det er midt i brugerens synsfelt. Siden kan scrolles, så man kan se alle klubber. Jeg havde fokus på, at man skulle have adgang til vigtige funktioner hele tiden selvom man havde scrollet. Derfor er der en menubar i toppen af skærmen, som er låst fast.

Det øvrige design har fokus på at være rart og indbydende. Næsten alting har derfor rundede hjørner, og der er benyttet blå som en gennemgående farve.

Produktets visuelle layout kan ses i test-afsnittet herefter.

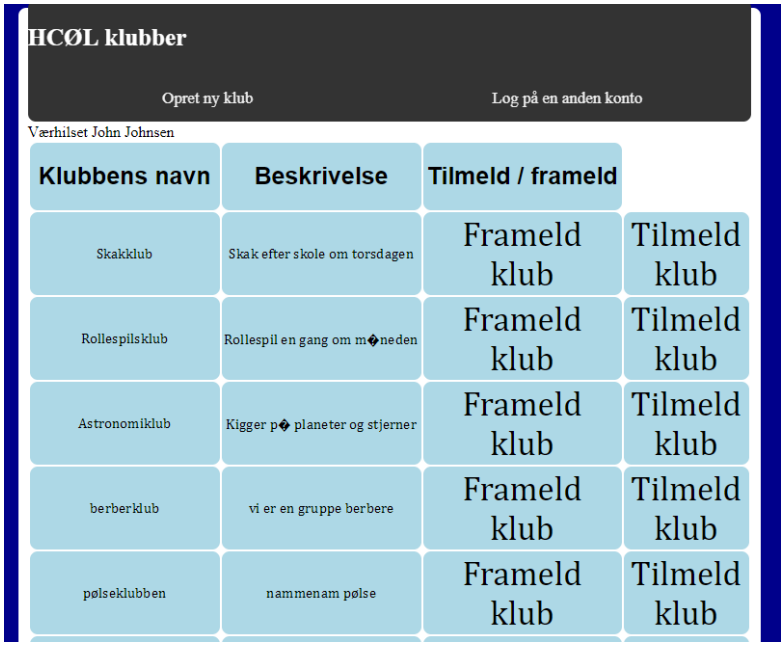
Test

For at undersøge om produktet virker efter hensigten, blev det testet, ved at lade som om man er en bruger som skal benytte produktets funktioner.

Først blev login-funktionaliteten testet som det kan ses på figur 11.

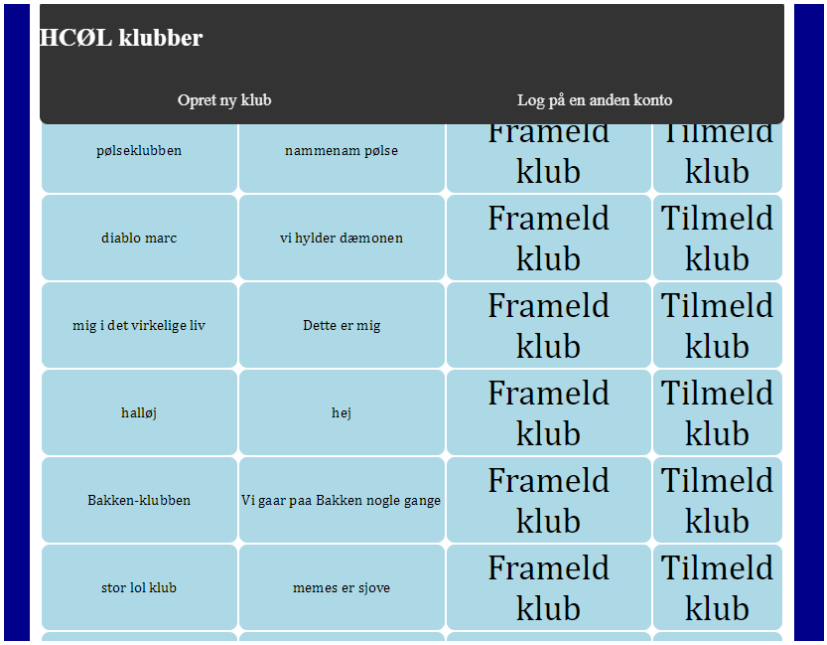
Figur 11: Login-siden (index.php).

Efter man er logget ind, kommer man til klubOversigt.php. Her kan det ses, at man bliver hilst med sit navn. Samtidig kan man se at klubberne bliver listet med deres tilhørende knapper. Dette kan ses på figur 12.



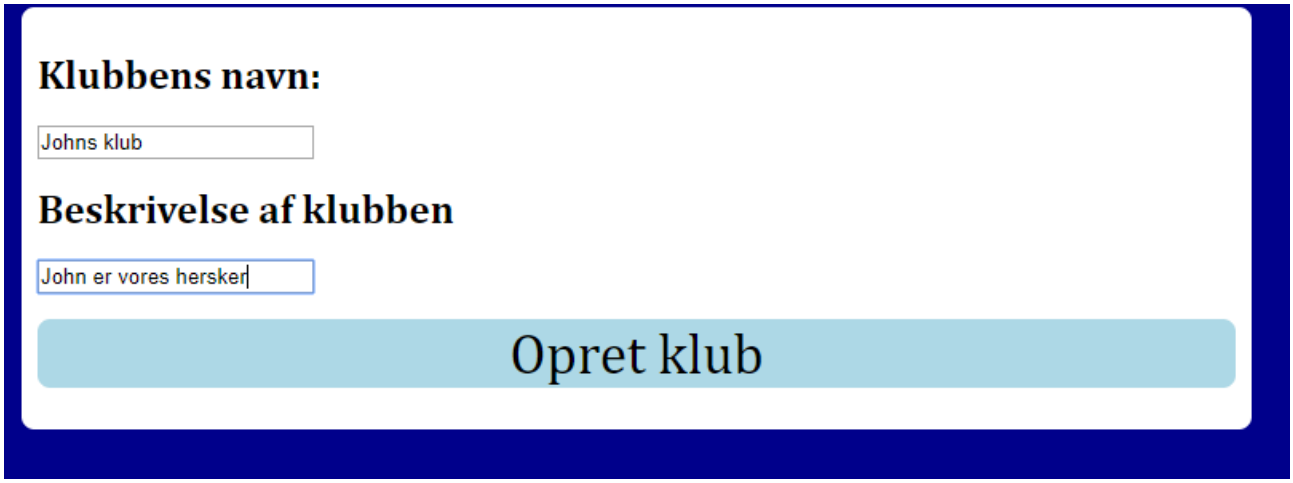
Figur 12: Hovedsiden (klubOversigt.php).

Nu testes scroll-funktionaliteten. Det testes altså om siden kan scrolle samtidig med at topbaren bliver hvor den er. Dette kan ses på figur 13.



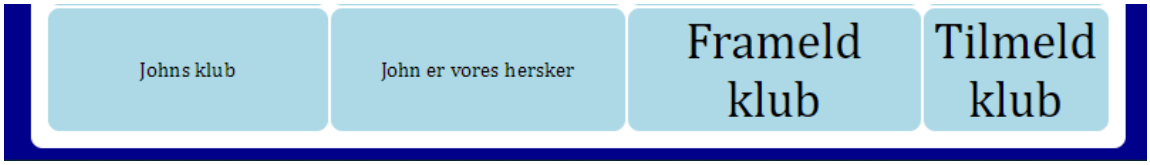
Figur 13: Man kan scrolle og topbaren bliver på plads.

Dernæst undersøges det om man kan oprette en klub. Der klikkes på ”Opret ny klub” øverst på skræmen, hvorved man bliver henvist til lavKlubSide.php. Her kan man indtaste klubbens navn og en beskrivelse af den, som det kan ses på figur 14.



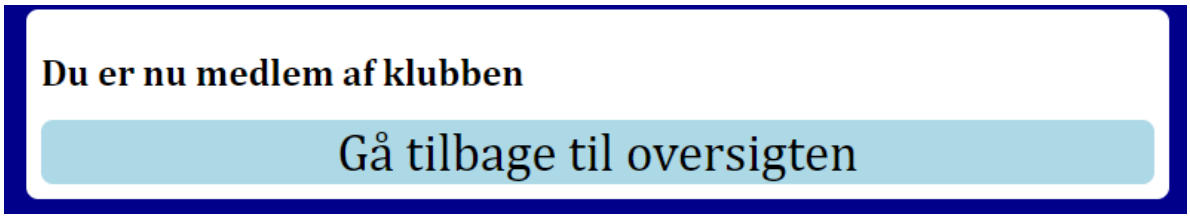
Figur 14: lavKlubSide.php hvor man kan oprette en klub.

Ved at klikke på ”Opret klub”, submittes de indtastede informationer til lavKlub.php. Det kan nu tjekkes om klubben er oprettet med succes, ved at kigge på hovedsiden. På figur 15 kan det ses at klubben er blevet oprettet.



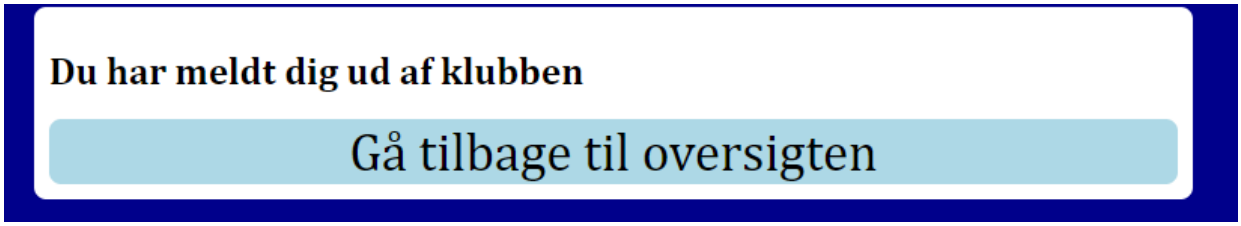
Figur 15: Klubben er blevet oprettet.

Nu forsøges at tilmelde en klub. Når der klikkes på ”Tilmeld klub” kaldes tilmeld-funktion så club_id og user_id insættes i userclub-tabellen. Herefter får man en godkendelse, som det kan ses på figur 16.



Figur 16: Det godkendes at man er medlem.

Ligeledes sker det når man melder sig fra. Her kaldes dog frameld-funktionen, og man får derefter en godkendelse, som det kan ses på figur 17.



Figur 17: Det godkendes at man har meldt sig ud.

Testene gik overordnet godt, da man med succes kan logge ind, oprette en klub og tilmelde sig klubber. Der blev dog fundet et problem med at opbevare session-værdier. Dette uddybes i afsnittet ”Eventuelle forbedringer”.

Sikkerhed

På grund af kodens opbygning, er det muligt at omgå nogle menuer ved at foretage SQL-injections. Dette vil sige at man kan indtaste værdier i URL-feltet, som vil blive sendt til databasen. Det menes dog at dette ikke er noget stort problem, da der ikke bliver behandlet nogle kritiske data. Det eneste man stort set kan foretage sig med en SQL-injection, er at melde sig ind eller ud af klubber. Hvis databasen behandlede kritiske data, for eksempel bankoverførsler, ville det være nødvendigt at øge sikkerheden. Både med hensyn til SQL-injections og den måde passwords bliver opbevaret. Som det er gjort i projektet, opbevares passwords i normal tekst i databasen. Hvis sikkerheden skulle højnes, kunne man benytte password hashing. Dette er en metode, hvorved kodeord bliver krypteret, således at hvis databasen lækker, vil man ikke kunne læse alle brugeres kodeord.

Et problem som produktet kan være udsat for når det offentliggøres, kan være DDoS-angreb. Dette står for ’distributed denial of service’, og det går ud på at en angriber bombarderer hjemmesiden med mange besøgende på en gang, så hjemmesiden går ned. Når det kommer til datasikkerhed, er dette ikke et problem, men det er et muligt problem i form af at hjemmesiden helst skulle være tilgængelig 24/7. Da produktet er målrettet skoleelever, menes det dog at der ikke vil være nogen fordel i at angribe hjemmesiden, og at der derfor nok ikke er nogen med evnerne til det som vil gøre det.

Eventuelle forbedringer

Der er nogle ting man kunne ændre ved koden, som muligvis ville gøre den bedre.

Tilføje if-statement

En af disse ting, kunne være at tilføje et if-statement, således at når man er medlem af en klub, vises kun ”afmeld”-knappen. En måde dette kunne gøres, er ved at tjekke ved hver klub om der i userclub-tabellen allerede er en sammensætning af den klub og brugeren. Dette ville se cirka ud som det gør på figur 18 herunder.

```
if($_SESSION['user_id']='user_id' AND $clubid='club_id'){
echo"
<td><a href='frameld.php?
club_id=".$row['id']."'>
<button>Frameld klub</button></a></td>";

}
else{
echo"
<td><a href='tilmeld.php?
club_id=".$row['id']."'>
<button>Tilmeld klub</button></a></td>";
}
```

Figur 18: Eksempel på hvordan et if-statement kunne se ud. Dette eksempel er ikke funktionelt, der skal kun kigges, på princippet i opbygningen.

Princippet i if-statementet, er at der tjekkes om brugerens user_id allerede findes kombineret med den givne klubs club_id i userclub-tabellen. Hvis det er tilfældet, vil kun ”afmeld”-knappen blive vist. Hvis ikke, vil kun ”tilmeld”-knappen blive vist.

Som programmet er nu, er det ikke et funktionelt problem at begge knapper vises, men det gør at en bruger kan melde sig en i den samme klub flere gange. Dette er ikke hensigtsmæssigt, men det er ikke noget stort problem, da frameld-funktionen vil fjerne alle entries i userclub-tabellen med den givne bruger-klub-kombination.

Spam-beskyttelse

Nogle brugere vil måske finde det sjovt at oprette rigtig mange ligegyldige klubber. Dette kunne undgås ved at lægge et loft over hvor mange klubber en bruger kan oprette. Hvis man til club-

tabellen tilføjede grundlæggerens user_id, kunne man gøre at den bruger så ikke må oprette flere klubber.

Elevernes opførsel vil bestemme om denne funktion er nødvendig. Generelt synes jeg at man skal have lov til at oprette mange klubber, hvis man gerne vil bidrage til skolens fællesskab. Så længe det ikke bliver misbrugt, er en sådan funktionalitet unødvendig.

Bedre opbevaring af session-værdier

Der er et større problem med produktets funktionalitet. Dette er, at brugerens session ikke bliver opbevaret ordentligt. Som det er i koden nu, vil man blive logget ind, og navnet vil være korrekt. Når man foretager en handling på siden, vil der dog ske en af to ting. Enten bliver man logget på en anden bruger, eller også får man en fejlmeddelelse. Dette er fordi session-værdierne ikke bliver opbevaret ordentligt. Hvis de blev opbevaret ordentligt, ville man forblive logget ind på den samme konto. Der er altså et sted i koden hvor session-værdierne går tabt og ikke bliver husket. Dette er et problem som skal løses før produktet er fuldt funktionelt. Princippet med produktet kan dog stadig fremvises på trods af denne fejl, så til demonstrationsformål er det ikke et stort problem.

Ekstra features

Ekstra features er ikke nødvendige for produktets grundfunktionalitet, men de kan gøre at produktet er brugbart i flere scenarier. En feature som med vilje er udeladt, er en knap til at fjerne klubber. Det blev bedømt at en sådan knap, ville skabe mere frustration for brugere end glæde, da nogle personer måske ville gå ind og slette klubber for sjov eller slette klubber ved en fejl.

Man kunne også have tilføjet en knap, som kunne vise hvem som er medlem af de enkelte klubber. Dette ville dog skabe et problem i form af brugernes privatliv. Nogle brugere ønsker muligvis ikke, at alle kan se hvilke klubber de er med i.

Vurdering og konklusion

Produktet stødte desværre ind i nogle funktionelle problemer undervejs. Dette hindrer det i at blive sat i værk som et funktionelt system. Hvis fejlen med session-værdier fikses, menes det dog at produktet vil være ganske brugbart og klart til brug på skoler. Produktet vil kunne dække et behov som der er, i form af den manglende orden over skolers klubber. Specifikt på HCØL, vil produktet kunne øge elevers viden om de forskellige klubber, og dermed højne det sociale sammenhold på tværs af klasser.

Datasikkerhed er dog et problem som er meget relevant i disse moderne tider. Viden om mennesker er en milliardindustri, som firmaer såvel som hackere meget gerne vil have fat i. Dertil kommer det, at når noget først er lækket og kommet på internettet, er det stort set umuligt at få væk igen. At øge sidens sikkerheds overfor bl.a. SQL-injections og password-tyveri, vil derfor være en topprioritet i en version 2.0 af produktet.

Produktet opfylder altså de specifikke krav som der blev opstillet, men videre udvikling af det, vil være hensigtsmæssigt før det sættes i funktion.

Bilag (kode)

database.php

```
<?php
    function forbindelse(){ //Funktion til at oprette forbindelse til SQL-
databasen.
        $connection = new mysqli("localhost", "root",""");//Indsæt eget kodeord
her.

        return $connection;
    }
    $connection = forbindelse(); //Opret forbindelse.

        $sql = "CREATE DATABASE IF NOT EXISTS clubDB"; //Opret databsen hvis den
ikke findes.
        $connection->query($sql);

//Hvis tabellerne ikke findes, oprettes de med automatisk ID som primary key.
$sql = "CREATE TABLE IF NOT EXISTS clubDB.CLUB (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, --ID tilegnes automatisk. Det er
primary key.
navn VARCHAR(30) NOT NULL, --Navnet må være på op til 30 tegn.
beskrivelse VARCHAR(100) NOT NULL)"; //Beskrivelsen må være på op til 100 tegn.

$connection->query($sql);
//Tabel til brugere oprettes på samme måde.
$sql = "CREATE TABLE IF NOT EXISTS clubDB.USERS (
        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        navn VARCHAR(30) NOT NULL,
        password VARCHAR(30) NOT NULL)";

$connection->query($sql);
//Tabel til at kombinere brugere og klubber oprettes.
$sql = "CREATE TABLE IF NOT EXISTS clubDB.USERCLUB (
user_id INT(6) UNSIGNED,
club_id INT(6) UNSIGNED,
--De er begge foreign keys, da de hentes fra andre tabeller.
FOREIGN KEY (user_id) REFERENCES clubDB.USERS(id),
FOREIGN KEY (club_id) REFERENCES clubDB.CLUB(id))";
$connection->query($sql);

    function opretBruger($connection, $navn, $password){ //Funktion til at
oprette brugere.
        $sql =  "INSERT INTO clubDB.USERS (navn, password) VALUES
('".$navn."','".$password."')"; //Navn og password indsættes i tabellen over
brugere.
        $userCreated = $connection->query($sql);
        return $userCreated;
    }

    function eksisterendeBruger($connection, $navn){ //Funktion til hvis
brugeren
        $sql = "SELECT * FROM clubDB.USERS WHERE navn='".$navn."' LIMIT 1";
        $result = $connection->query($sql);
        $row = $result->fetch_assoc();

        return $row!=null;
```

```
    }

    function rigtigKode($connection, $navn, $password){ //Funktion som kan
tjekke om en bruger findes i forvejen eller om der er tale om en ny bruger.
        $sql = "SELECT USERS.id FROM clubDB.USERS WHERE password='".$password."'
AND navn='".$navn."' LIMIT 1";
        $result = $connection->query($sql);
        $row = $result->fetch_assoc();
        return $row != null ? $row['id']: null;
    }

    //Når denne funktion kaldes, fjernes en entry fra userclub hvor der er en
specifik kombination af user_id og club_id
    function frameld($connection, $userid, $clubid){
        $sql = "DELETE FROM clubDB.userclub WHERE
user_id=".$_SESSION['user_id']." AND club_id=".$clubid."";

$tbCreated = $connection->query($sql);
    }

    //Når denne kaldes indsættes der i userclub.
    function tilmeld($connection, $userid, $clubid){
        $sql ="INSERT INTO clubDB.userclub (`user_id`, `club_id`) VALUES
(".$_SESSION['user_id']."., ".$clubid.");";

        $tbCreated = $connection->query($sql);
        }
?>
```

index.php

```
<html>
<title>Log ind</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
<body>
<?php

?>
<div class="content">
<h2>Log ind og meld dig til klubber</h2>
<h5>Hvis du er ny her, så skriv dit navn og et kodeord, så bliver du
oprettet</h5>
<!--Der indtastes navn og kodeord. Dette postes ind i klubOversigt.php. -->
<form action="klubOversigt.php" method="post">
    <h1>Navn:</h1>       <input type="text" name="navn" placeholder="Navn"><br>
    <h1>Password:</h1>   <input type="password" name="password"
placeholder="Kode"><br>
    <p>
        <button input type="submit">Opret eller login</button>
    </p>
</form>
    </div>
</body>
</html>
```

klubOversigt.php

```
<html>
    <head>
        <title>HCØL klubber</title>
        <link rel="stylesheet" type="text/css" href="styles.css">
    </head>
<body>
    <div class="contentMain">

<!--Dette er den øverste menubar som er låst fast. -->
<div class="topBar">
    <h3>HCØL klubber</h3>
  <a href="lavKlubSide.php">Opret ny klub</a>
  <a href="index.php">Log på en anden konto</a>

</div>

<?php
    session_start();
    include "database.php";
    mysqli_select_db($connection,'clubDB');
    $navn  = isset($_POST['navn']) ? $_POST['navn'] : $_SESSION['navn'];
//Brugerens navn sættes.
    $password = isset($_POST['password']) ? $_POST['password'] :
$_SESSION['password']; //Brugerens kodeord sættes.

    $connection = forbindelse();
    $brugerEksisterer = false;

    if($brugerEksisterer){ //Hvis brugeren allerde findes, tages vedkomnes
værdier og sættes ind.
        $_SESSION['navn'] = $navn;
        $_SESSION['password'] = $password;
        $_SESSION['connection'] = $connection;
        $_SESSION['user_id'] = rigtigKode($connection,$navn,$password);
        }

    if(rigtigKode($connection,$navn,$password)){ //Hvis brugeren findes og koden
er rigtig.
        echo "Værhilset ".$navn;
        $brugerEksisterer   = true;
    }else if (eksisterendeBruger($connection,$navn)){ //Hvis koden er skrevet
forkert.
        echo "Forkert kodeord, ikke logget ind";
        $brugerEksisterer   = false;
    }else{ //Hvis det er en ny bruger.
        echo "Bruger oprettet";
        $brugerEksisterer   = opretBruger($connection,$navn,$password);
    }


    mysqli_select_db($connection,'clubDB'); //Databasen vælges.

$query = "SELECT * FROM club"; //Tabellen vælges.
$result = mysqli_query($connection,$query);

//Tabellen startes i HTML. <th> sætter overskrifter ind.
echo "<table>
```

```
<th>Klubbens navn</th>
<th>Beskrivelse</th>
<th>Tilmeld / frameld</th>";
//While loop som kører det antal gange der er rækker.
while($row = mysqli_fetch_array($result)){
echo "<tr>
<td>" . $row['navn'] . "</td><td>" . $row['beskrivelse'] . "</td>


<td><a href='frameld.php?
club_id=".$row['id']."'>
<button>Frameld klub</button></a></td>
<td><a href='tilmeld.php?
club_id=".$row['id']."'>
<button>Tilmeld klub</button></a></td>
</tr>"; //Frameld og tilmeld knapper sættes ind. De dirigerer henholdsvis til
frameld.php og tilmeld.php
}
echo "</table>";

            ?>
        </div>
    </body>
</html>
```

tilmeld.php

```
<head>
    <title>Tilmeld</title>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>

<body>
    <div class="content">
<?php
    session_start();
    include "database.php";
    $connection = forbindelse();
    tilmeld($connection, $_SESSION['user_id'], $_GET['club_id']);//Funktionen
kaldes fra database.php og der indsættes sessionens user_id samt det givne
club_id.

    echo "<h1>Du er nu medlem af klubben</h1>";
    echo "<a href='klubOversigt.php'><button>Gå tilbage til
oversigten</button></a>";//Knap til at vende tilbage til hovedsiden.
    $tbCreated = $connection->query($sql);


?>
    </div>
</body>
```

frameld.php

```
<head>
```

```
    <title>Frameld</title>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>

<body>
    <div class="content">
<?php
    session_start();
    include "database.php";
    $connection = forbindelse();
    //Der benyttes sessionens user_id og det club_id som blev givet af URL'en.
    frameld($connection, $_SESSION['user_id'], $_GET['club_id']);

    echo "<h1>Du har meldt dig ud af klubben</h1>";
    //Knap til at vende tilbage til hovedsiden.
    echo "<a href='klubOversigt.php'><button>Gå tilbage til
oversigten</button></a>";
    $tbCreated = $connection->query($sql);
?>
    </div>
</body>
```

lavKlubSide.php

```
<html>
<head>
    <title>Opret klub</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
    <body>
        <div class="content">
        <!--Værdierne postes ind i lavKlub.php -->
        <form action="lavKlub.php" method="post">
        <h1>Klubbens navn:</h1>       <input type="text" name="navn"><br>
        <h1>Beskrivelse af klubben</h1>   <input type="text"
name="beskrivelse"><br>
        <p><!--p benyttes til at rykke knappen lidt længere ned.-->
        <button input type="submit">Opret klub</button>
        </p>
        </form>
        </div>
    </body>
</html>
```

lavKlub.php

```
<?php
    session_start();
    include "database.php";
    $navn = $_POST['navn']; //Navnet er det som blev postet fra lavKlubSide.php
    $beskrivelse  = $_POST['beskrivelse']; //Beskrivelsen fås også derfra.
    $connection = forbindelse();
    //I club-tabellen indsættes værdierne i kolonnerne navn og beskrivelse.
    $sql ="INSERT INTO clubDB.CLUB (`navn`, `beskrivelse`) VALUES
('".$navn."','".$beskrivelse."');";
```

```
    $tbCreated = $connection->query($sql);
        //Der dirigeres tilbage til klubOversigt.php.
        if ($tbCreated) {
             header ("Location:klubOversigt.php");

}
?>
```

styles.css

```
/*Her defineres skrifttyperne*/
h1 {
    color: black;
    font-size: 25px;
    font-family: Cambria, "Hoefler Text", "Liberation Serif", Times, "Times New
Roman", "serif";
}

h3 {
    color: white;
    font-size: 25px;
}
h6 {
    color: white;
    font-size: 25px;
    text-align: right;
}
/*Overskrifterne i tabellen sættes her*/
th {
font-size: 25px;
}
/*Teksten inde i tabellen sættes her*/
td {
    color: black;
    font-size: 15px;
    font-family: Cambria, "Hoefler Text", "Liberation Serif", Times, "Times New
Roman", "serif";
}
/*Hvordan selve tabellen skal se ud sættes her*/
table {
    font-family: arial, sans-serif;
    border-collapse: separate;
    width: 100%;
    height: 200px;
}
/*Ting som er gældende for både overskrifterne og teksten inde i tabellen*/
td, th {
    text-align: center;
    border-radius: 8px;
    width: 33.3%;
    background-color: lightblue;
    height: 70px;
}
/*Knappernes udseende defineres*/
button {
    border-radius: 8px;
```

```
    width: 100%;
    border: none;
    text-decoration: none;
    cursor: pointer;
    background-color: lightblue;
    color: black;
    font-size: 35px;
    font-family: Cambria, "Hoefler Text", "Liberation Serif", Times, "Times New
Roman", "serif";

}
/*Hvordan knapper ser ud når man holder over dem*/
button:hover{
background: black;
 color: white;
}
/*Baggrundens farve i siderne af hjemmesiden*/
body {
    background-color: darkblue;
}
/*class som sætter ting i midten og definerer bredden*/
div.content {
    max-width: 750px;
    margin: auto;
    background: white;
    padding: 10px;
    border-radius: 8px;
}
/*Denne class benyttes kun til klubOversigt.php, da der var behov for mere
padding i toppen*/
div.contentMain {
    max-width: 750px;
    margin: auto;
    background: white;
    padding: 10px;
    padding-top: 120px;
    border-radius: 8px;
}
/*Hvordan menuen øverst skal se ud*/
.topBar {
overflow: hidden;
background-color: #333;
position: fixed;
top: 0;
width: 750px;
border-radius: 8px;
}

.topBar a {
float: left;
display: block;
color: #f2f2f2;
text-align: center;
padding: 14px 16px;
text-decoration: none;
font-size: 17px;
width: 45.5%;
}
```

```
.topBar a:hover {
background: #ddd;
color: black;
}
```