



SKRIFTLIG EKSAMEN

Jonas Urup & Jacob Skadborg

Navn: _____

Informationsteknologi B

Fag: _____

2.T

Klasse: _____

26/4-2018

Dato: _____

Jeg bekræfter herved med min underskrift, at opgavebesvarelsen er udarbejdet af mig. Jeg har ikke anvendt tidligere bedømt arbejde uden henvisning hertil, og opgavebesvarelsen er udfærdiget uden anvendelse af uretmæssig hjælp og uden brug af hjælpemidler, der ikke har været tilladt under prøven

Jacob Skadborg

Jonas Urup Henriksen

Underskrift - elev

Opgaven er modtaget af:

Underskrift

DOKUMENTATION – OVERVÅGNING

TITELBLAD

- Titel – Dokumentation - Overvågning
- Tema - Overvågning
- Projektperiode - Forventet periode: 21-3-2018 til 26-4-2018
- Sidetal - 15 sider
- Forfattere – Jonas Urup Henriksen & Jacob Skadborg
- Skolens navn – Vejle Teknisk Gymnasium
- Klasse – 2.T
- Semester og år – 4. semester, 2. år
- Vejleder – Torben Holst
- Summary

In this exam project we have tried to make a service where people can order food from their local bakers easily online. The idea is that it will reduce food waste since the bakers will know exactly how much bread he must make each day. In this document you will get a look at how we made the side and our thought process leading us to the finished product.

INDHOLDSFORTEGNELSE

Titelblad	1
Forord	3
Projektbeskrivelse	3
Trinvis-udvikling	3
Tidsplan	4
Målgruppe	4
Produktudvikling	5
Designfase	6
Koden forklaret	8
Bestilling.php	8
Google Maps API	9
Databasen	10
Registrer.php	11
Log_ind.PHP	12
Alle sider	12
Flowchart over login system	12
Test af website	14
Konklusion	14
Evaluerings	15
Referencer	15

FORORD

I dette projekt har vi taget udgangspunkt i temaet overvågning. For at kunne implementere noget overvågning som ikke bare er ondsindet som folk hurtigt kan misforstå det til at være, har vi også prøvet at mindske madspild undervejs.

Vi har haft 12 uddannelsestimer og 3 elevtimer til projektet. Til hver af vores 2 gruppemedlemmer. Vi har haft en vejleder til rådighed i alle uddannelsestimerne.

Undervejs i rapporten har vi brugt billeder til at illustrere og supplere vores tekst. Der vil stå en figur tekst under hvert billede som forklare det. Hvis man har lyst til at kigge på koden som filen ankom sammen med, så er man meget velkommen, der vil være et par kommentarer dertil.

Vil du logge ind på hjemmesiden skal du bruge følgende adgangsoplysninger:

E-mail	Adgangskode
j.skadborg00@gmail.com	123
Jonas-urup-henriksen@hotmail.com	321

PROJEKTBESKRIVELSE

I Danmark findes der måder som sørger for at restauranter og andre madservices kan komme af med nogle af deres madrester, ligesom fx. Too Good To Go, hvor folk kan købe billige rester sidst på aftenen. Men desværre er der stadig store mængder madspild, specielt i bage- og kagebranchen, derfor vil vi gentænke måden at forhindre madspild, ved frem for at komme efter problemet når maden allerede er skabt, vil vi sørge for at vi ikke skaber for meget mad på første hånd. Det vil vi gøre ved at lave en hjemmeside/applikation hvor man som bruger kan bestille det brød man vil have ved bageren, og så kan bageren forberede hvor meget brød de vil lave.

Vi vil have at man skal kunne trykke sig rundt inde på siden ved hjælp af interaktion for at kunne finde de bagere der skal bestilles brød ved.

Vi vil have et login system som gør at vi kan få nogle informationer ud af brugeren af siden da projektet omhandler overvågning. Vi vil derudover også gerne se om vi kan implementere et Google Maps API så man kan tracke brugerens lokalisation.

TRINVIS-UDVIKLING

I trinvis-udvikling går det meget ud på at sætte nogle starts kravspecifikationer op og lave opgaven mere struktureret og sværere efterhånden som vi udvikler det så vi har derfor startet ud med at sige at vores side skal indeholde et simpelt system for at man ikke bliver forvirret og ikke kan benytte sig af siden

Vi skal derfor først have lavet et design, baseret på vores målgruppeanalyse. Derefter vil vi skrive det basale kode, så vi har et godt grundlag, og har fået vores hjemmeside til at se ud som den skal. Derefter udvider vi vores hjemmesides funktionalitet, så den bliver mere raffineret. Til sidst, når alle elementerne på vores hjemmeside er implementeret, vil vi omstrukturere for at få vores side til at se mere professionel ud.

PRODUKTUDVIKLING

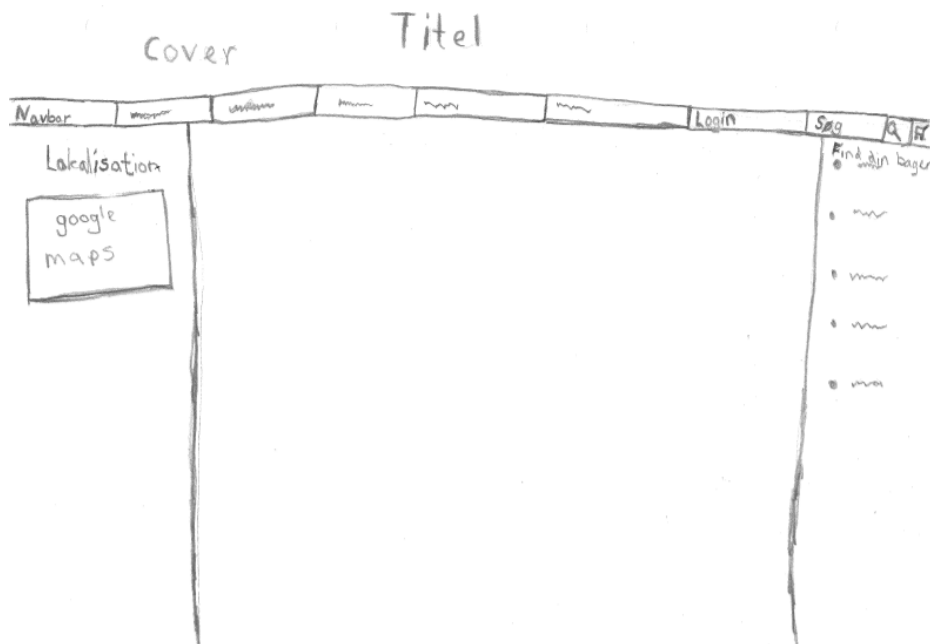
Vi synes at vi mangler en god hjemmeside til at kunne bestille sin egen mad eksempelvis i en bager og vi har derfor taget udgangspunkt i at det er en bager vi har lavet hjemmesiden til. Vi har været inde og kigge på nogle forskellige hjemmesider hvor man kan bestille brød og kager online men der er ikke nogen af dem der har set pæne og overskuelige ud og vi vil derfor lave en ny en som har begge disse ting eksempel på en af disse sider kan ses på figur 2. Vores side kunne sagtens blive omkonstrueret til at fungere i et normalt supermarked også men layout og overskuelighed skulle laves bedre end de allerede eksisterende.



Figur 2 Eksempel på en side som ikke hare været tilfredsstillende og som vi har prøvet at forbedre.

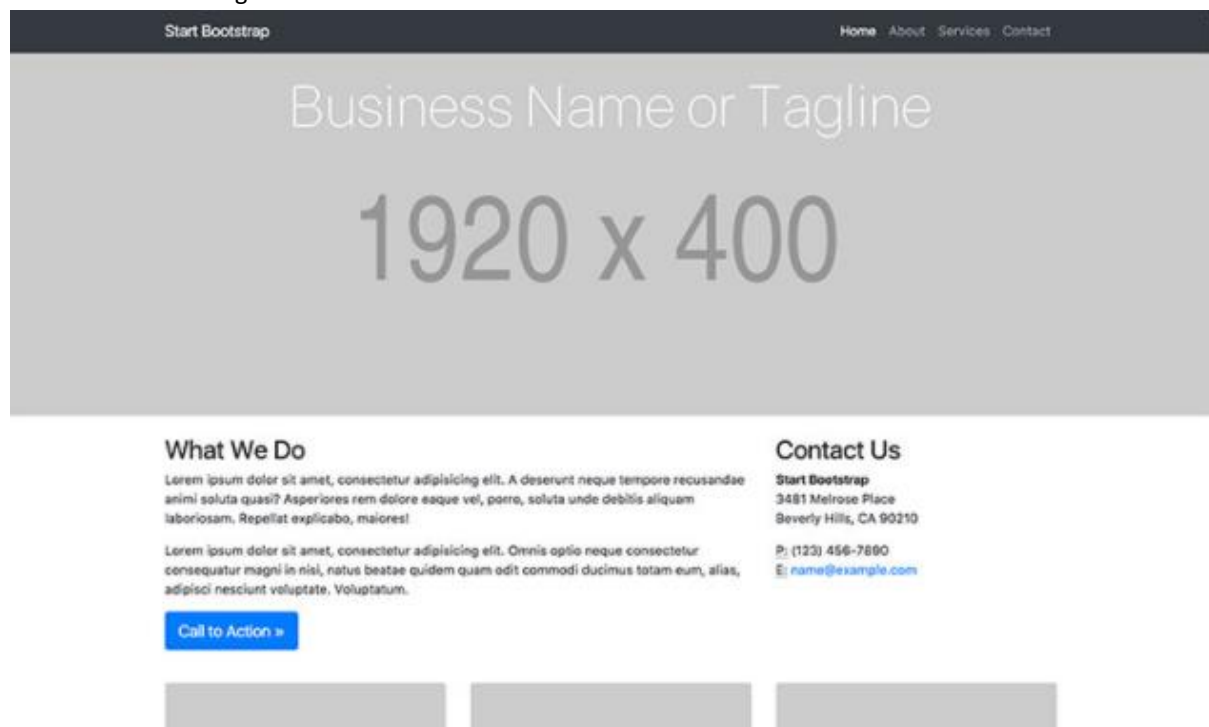
DESIGNFASE

I vores design vil vi gå ud fra vores målgruppeanalyse for at få et design som er overskueligt for de brugere vi søger. I vores målgruppeanalyse fandt vi ud af at vi skulle gå ud fra at det er mennesker i 18-65 års alderen både mand og kvinde da det er dem der handler mest ind og kan overskue at bestille online. Vi har startet med at lave en skitse at gå ud fra som man kan se på figur 3.



Figur 3 Første skitse over hvordan vores side skal være sat op

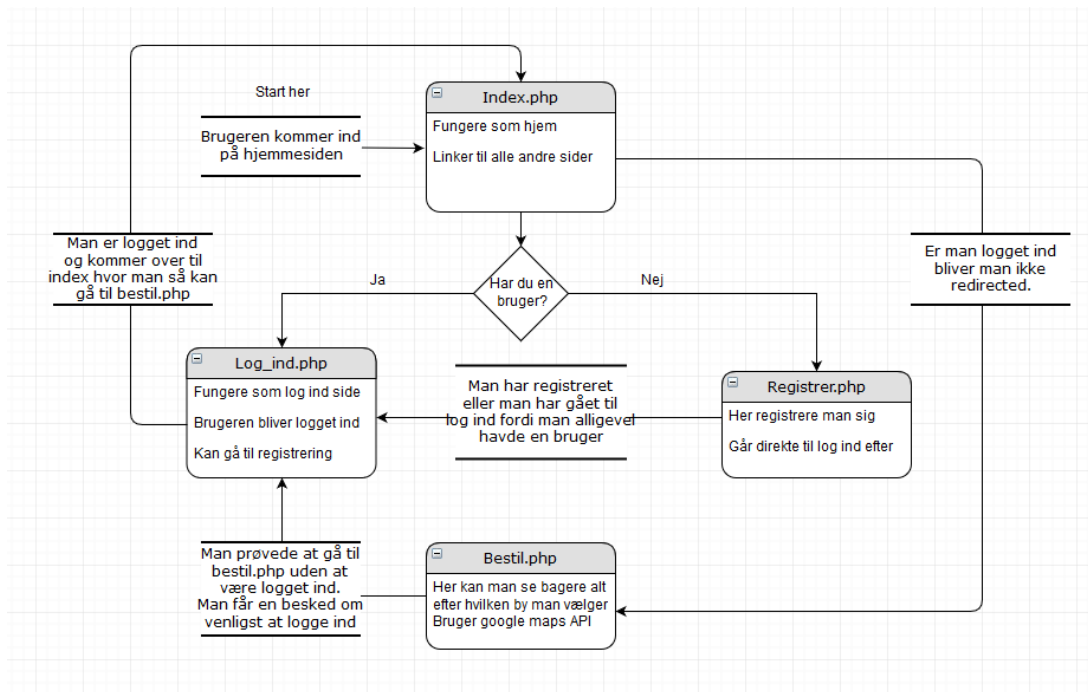
Vi har senere været inde på Bootstrap for at kigge efter et design til at starte med så vi ikke behøvede at lave hele hjemmesiden fra bunden. Vi har så fundet det design på Bootstrap der mindede mest om vores skitse og som så bedst ud se figur 4.



Figur 4 Template som har hjulpet os i gang med Bootstrap.

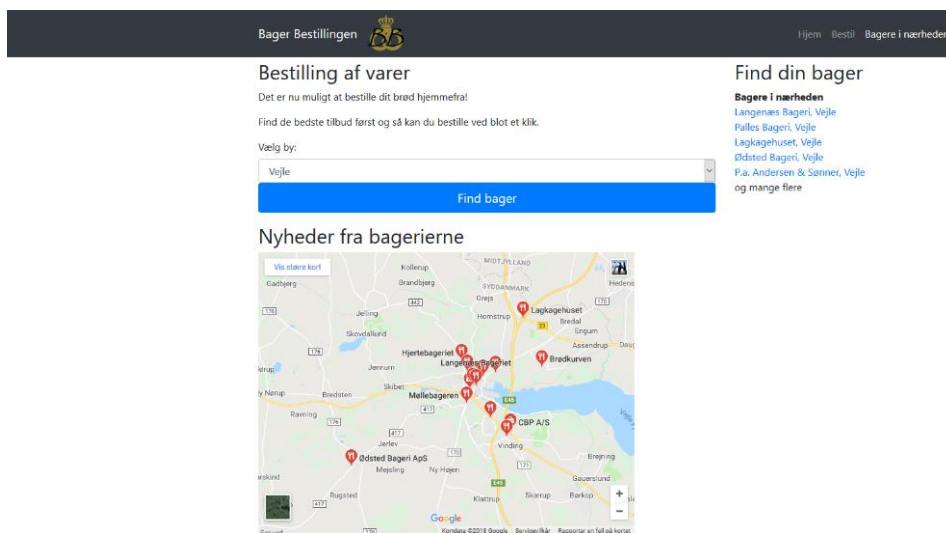
Efter vi havde fået fat i et pænt design, kunne vi begynde at redigere på det som at skifte baggrunden ud og lave helt om på tekst og knapper. Det vi har lagt meget vægt på er at få selvfølgelig også have noget at gøre med overremnet overvågning. Ved at lave et login system kan vi inde på vores database lagre brugerens navn, efternavn, adgangskode og e-mail og allerede der er det nok til at det er overvågning.

Loginsystemet er lavet ved brug af en database som er forbundet med vores hjemmesides login system så den tager oplysningerne der bliver skrevet ind i tekstfeltet og sender dem videre over i databasen og lagre dem op.



Figur 5 Sitemap som viser hvordan vores hjemmesider er sat op

Vi har på skitsen også tænkt at brugeren nok gerne vil finde den tætteste bager i nærheden af dem for hurtigst afhentning og vi har derfor også valgt at vi vil implementere en Google Maps API hvor du kan vælge din lokalisering. Vi kan på figur 7. se hvordan Google Maps er implementeret på vores hjemmeside. For at lave google Maps fik vi brug for en Google Maps API som er lavet af Google selv.



Figur 6 Billede af google Maps lige efter det blev implementeret

BESTILLING.PHP

For at vores hjemmeside blev mere interaktiv vil vi gerne have at brugeren kunne vælge hvilken by han gerne vil se bagere fra. For at kunne gøre dette var vi nødt til at opsætte en form med valgmuligheder, se linje 59-70 på figur 8. Vi skal derefter kunne finde ud af, hvilken by vores bruger har valgt, til det skal vi lave noget PHP som skal echo forskelligt html kode ud alt efter hvilken by de har valgt. Efter vi har fundet ud af hvilken by de har valgt, skal vi lave betingelser til hver af byerne. Det gør vi med if statements. Vi tjekker først om der er blevet trykket på submit knappen, hvis der ikke er, vil den vise Vejle bagere som standart. Det har vi gjort for at listen ikke bare skal være tom når man lige er kommet ind på siden. I den samme if statement tjekker man så hvilken by der er valgt. Det gør vi ved at skrive && som betyder OG. Så hvis der er trykket på knappen OG man har valgt Give, vil linje 120 på figur 8 være korrekt og listen som går derunder vil derfor blive vist.

For at kunne vise forskelligt html på din hjemmeside efter behov kan man bruge echo funktionen fra php hvori man kan skrive sin html uden at ændre på html teksten.

I eksemplet vil man have *Vejle+bager* som tekst i html

```
if (!isset($_POST['submit'])) {
    echo "Vejle+bager";
}
```

Figur 7 Eksempel på hvordan man kan bruge php sammen med html

```
59 <div class="form-group">
60 <form method="post">
61 <label for="liste">Vælg by:</label>
62 <select class="form-control" id="liste" name="by">
63 <option value="Vejle">Vejle</option>
64 <option value="Jelling">Jelling</option>
65 <option value="Bredsten">Bredsten</option>
66 <option value="Give">Give</option>
67 </select>
68 <button class="btn btn-lg btn-primary btn-block" id="submit" type="submit" name="submit">Find bager</button>
69 </form>
70 </div>
71
72 if (!isset($_POST['submit'])) {
73     echo "
74     <address>
75     <strong>Bagere i nærheden</strong>
76     <br><a href='\"http://www.langenaesbageriet.dk/langenaes-bageriet-vejle/\">Langenæs Bageri, Vejle</a>
77     <br><a href='\"http://www.pallesbageri.dk/index.php?sideid=103&hovedsid=10\">Palles Bageri, Vejle</a>
78     <br><a href='\"https://lagkagehuset.dk/butikker\">Lagkagehuset, Vejle</a>
79     <br><a href='\"https://lagkagehuset.dk/butikker\">Ødsted Bageri, Vejle</a>
80     <br><a href='\"http://www.vindinggaardcenter.dk/PaKringleriet.html\">P.a. Andersen & Sønner, Vejle</a>
81     <br>og mange flere
82     </address>";
83 }
84
85 if (isset($_POST['submit']) && $_POST['by'] == 'Give') {
86     echo "
87     <address>
88     <strong>Bagere i nærheden</strong>
89     <br>Nytors Bageri, Give
90     <br>Superbrugsen, Give
91     </address>";
92 }
```

Figur 8 Kode fra bestil.php. Viser hvordan vi har lavet vores side interaktiv så brugeren kan vælge forskellige byer. Læg mærke til at linjetallene hopper, så koden er klippet sammen for at vise det mest relevante.

GOOGLE MAPS API

For at give vores brugere en endnu bedre oplevelse, hvor de kan få overblik over hvilke bagere der er i de forskellige byer, besluttede vi os for at bruge Google Maps Embed API til at embede en søgning på Google Maps i vores hjemmeside.

For at sige det simpelt er en API en måde at tilbyde en tjeneste fra et system, som folk så kan bruge til et andet system. Det er funktioner, ligesom dem man selv ville skrive, som er offentliggjort så andre kan bruge dem.

Embedding betyder at vi tage en del af en anden hjemmeside og propper den ind midt i vores egen. Det er noget der ofte bliver brugt på hjemmesider. En hjemmeside som linker til deres egen Twitter profil for eksempel bruger en del af Twitters hjemmeside midt i deres egen for at vise logoet.

Google har fantastisk dokumentation hvad dette angår, og hvis man ved hvor man skal kigge, tager det ikke så lang tid at finde ud af. Deres API lader os placere et interaktivt kort på vores side ved at lave en HTTP request med HTML tagget `<src>`.

På deres hjemmeside har de opsat en template så man hurtigt kan komme i gang med deres API, se figur 10. Man skal bruge en API key fra dem fordi så kan de se, at det er en der har fået lov, og de kan dermed også se *hvem* det er der bruger den, til hvad. Med en API key kan google se, hvad du bruger deres API til, altså hvilke requests du laver.

I vores tilfælde har vi ikke noget problem med at holde os sikre da det ikke ville blive offentliggjort, men under andre forhold ville man holde sin API key privat for at andre ikke skal kunne bruge den. På figur 10 kan man se et `&q=` på linje 6 som er her vores søgning skal placeres. Fx ville vi skrive `&q=Vejle+bager` for at søge efter Vejle bager

```
<iframe
  width="600"
  height="450"
  frameborder="0" style="border:0"
  src="https://www.google.com/maps/embed/v1/place?key=YOUR_API_KEY
    &q=Space+Needle,Seattle+WA" allowfullscreen>
</iframe>
```

Figur 10 Googles template som vi har brugt til at embedde en Google Maps søgning på vores side.



Figur 9 Et eksempel på hvor ofte folk embedder ting på deres hjemmesider. Taget fra TheLazyGoldmaker.com

IMPLEMENTERING

Når vi har embedded Google Maps på vores side kan vi lave en bestemt søgning, og så vise den. Men vi vil gerne have at hver by skal søges efter når man vælger den. Det kræver at vi inde i den template Google har lavet, bruger PHP til at skrive forskellige søgninger. Ligesom med vores automatiske tabel på den selv samme side bruger vi if statements til hver by, og igen vil vi have en søgning til hvis der ikke er blevet sat en bestemt by.

Som man kan se på figur 11, vil vi, alt efter hvilken if statement der er korrekt, skrive de forskellige tekste ved *echo*, så man søger efter det, som er blevet valgt fra vores form. Sammenlign figur 10 og figur 11.



Figur 11 Viser hvordan vi har brugt PHP til at indsætte forskellige tekster i vores google request alt efter hvilken by man har valgt. Se figur 10 for sammenligning mellem de to <src> tags.

DATABASEN


Vi har brugt database for at kunne lagre information på siden som nævnt før. Måden det er gjort på er en af de letteste måder at gøre det på som er ved hjælp af MySQL som er en kæmpe database online. Ved hjælp af MySQL kan vi submitte data fra brugeren over på en database som brugeren ikke kan se og derved lagre informationen og genbruge det til fx en registrering og login funktion som på vores hjemmeside. Måden vi har forbundet til serveren, er ved at lave en ny server på phpMyAdmin med navne som vi kan fremkalde i vores connect.php. Se figur 12 for reference.

Connect.php åbner egentlig bare op for den server vi har fortalt den om fra userinformation.php. Når begge disse filer er lavet, er vi i princippet forbundet med databasen når vi har inkluderet det i den fil/filer vi skal bruge den som i vores tilfælde er registrer.php og log_ind.php.

De informationer vi har hentet fra vores side og lagt over på databasen, er fornavn, efternavn, e-mail og adgangskode fra brugeren. På figur 13 ses hvordan det er sat op.

```
$host_db = "localhost";
$user_db = "root";
$password_db = "root";
$database_db = "nootnoot";
    $table1_db = "table1";
    $table2_db = "blog";
    $table3_db = "images_tbl";
    $table4_db = "questions";
    $table5_db = "eksamen_users";
```

Figur 12 Vi bruger de første 4 linjer til at forbinde med vores connect.php og derefter bruges \$table5_db da det er der vores selvvalgte informationer fra databasen er.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	id 	int(11)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	email	text	latin1_swedish_ci		No	None		
<input type="checkbox"/> 3	password	text	latin1_swedish_ci		No	None		
<input type="checkbox"/> 4	fornavn	text	latin1_swedish_ci		No	None		
<input type="checkbox"/> 5	efternavn	text	latin1_swedish_ci		No	None		

Figur 13 Vores database inde på phpMyAdmin med de informationer vi vil have fra brugeren.

Vi har også et id inde i databasen som vi bliver nødt til at have for at hver bruger er unik hvilket man også kan se ved AUTO_INCREMENT da det gør at først bruger kommer til at hedde id=1, næste bruger id=2 osv.

REGISTRER.PHP

Nu hvor databasen er sat op kan vi begynde at arbejde i vores registrer.php og få værdierne ind i MySQLi og det gør vi struktureret ved at give vores superglobals en værdi så det er lettere at finde rundt i og indsætte i MySQLi som kan ses på figur 14.

```

5  if (isset($_POST['submit'])) {
6      $email = $_POST['inputEmail'];
7      $password = $_POST['inputPassword'];
8      $fornavn = $_POST['inputFornavn'];
9      $efternavn = $_POST['inputEfternavn'];
10
11     $mysqli->query("INSERT INTO $table5_db (email, password, fornavn, efternavn) VALUES ('$email', '$password', '$fornavn', '$efternavn')");
12 }
13
14 $mysqli->close();

```

Figur 14 Superglobals får en værdi så de er lettere at arbejde med fra linje 6-9 og derefter sat i mysqli på linje 11.

Vi kan nu lave et formfelt så brugeren har en måde at skrive deres informationer ind i databasen på. Formfeltet er på vores side de felter man skriver i når man skal registrere sin bruger.

```

71 <form class="form-signin" method="post">
72 <h2 class="form-signin-heading mt-4">Registrer</h2>
73 <label for="inputFornavn" class="sr-only">Fornavn</label>
74 <input type="text" id="inputFornavn" class="form-control" placeholder="Fornavn" name="inputFornavn" required>
75 <label for="inputEfternavn" class="sr-only">Efternavn</label>
76 <input type="text" id="inputEfternavn" class="form-control" placeholder="Efternavn" name="inputEfternavn" required>
77 <label for="inputEmail" class="sr-only">Email adresse</label>
78 <input type="email" id="inputEmail" class="form-control" placeholder="Email adresse" name="inputEmail" required autofocus>
79 <label for="inputPassword" class="sr-only">Adgangskode</label>
80 <input type="password" id="inputPassword" class="form-control" placeholder="Adgangskode" name="inputPassword" required>
81 <button class="btn btn-lg btn-primary btn-block" id="submit" type="submit" name="submit">Registrer</button>
82 </form>
83 <p>Har du allerede en bruger? Så log ind </p>
84 <a href="log_ind.php" class="btn btn-default btn-lg active" role="button">Log ind</a>
85 </div>
86 </div>

```

Figur 15 Formfeltet brugeren skriver deres informationer ind i for at registrere en ny bruger.

Formfeltet bruger også værdierne fra figur 14. da det er de informationer vi vil have ind i databasen. Hvis vi ikke har genkaldt navnene som kan ses på bl.a. linje 74 med inputFornavn vil der ikke ske noget når brugeren vil registrere da deres nedskrevne information ikke vil blive sendt videre til databasen. Efter alle disse ting var kommet på plads uden fejl i koden, kunne vi starte på log_ind.php. Log_ind.php er også linket på linje 84 hvis brugeren allerede har registreret. Dette er udelukkende for brugervenligheden da det gør det lettere og hurtigere for brugeren at komme videre til log_in.php.

LOG_IND.PHP

Når vores bruger har registreret sig, har vi gemt deres information på vores database. For at de så skal kunne logge ind, skal vi finde en måde hvorpå vi kan sammenligne e-mailen som de skriver i log in feltet med den e-mail vi har gemt i databasen. Hvis vi kan se, at e-mailen er korrekt, altså at den samme e-mail som de har skrevet, er registreret i vores database, kan vi finde ud af hvilken række e-mailen er på og så om passwordet stemmer overens med det brugeren har skrevet ind. Hvis de kriterier er mødt, vil brugeren have skrevet rigtige log ind oplysninger ind, og et session navn vil blive sat.

Vi vil ikke have at man skal kunne på ind og se bagere i nærheden uden at logge ind, og det undgår vi ved at videresende folk der ikke har et session navn til log_ind.php, se figur 16.

```
3 if($_SESSION['login_user'] == ""){
4     header("Location: log_ind.php?info=nologin");
5 }
```

Figur 16 linje 3-5 i bestil.php. Viser hvordan vi smider folk ud hvis de ikke er logget ind

ALLE SIDER

Øverst på alle vores sider har vi startet en session. Det har vi gjort for at man ved bestil.php kan tjekke om den session så har værdien login_user hvilket kun er givet til en, hvis man har logget ind.

```
24 if($count == 1) {
25     $_SESSION['login_user'] = $myemail;
26     header("Location: index.php?info=login");
27     //Man er logget ind.
28 }
```

```
1 <?php
2 session_start();
```

Figur 17 linje 1 og 2 i index.php. Viser at vi har startet en session på alle siderne.

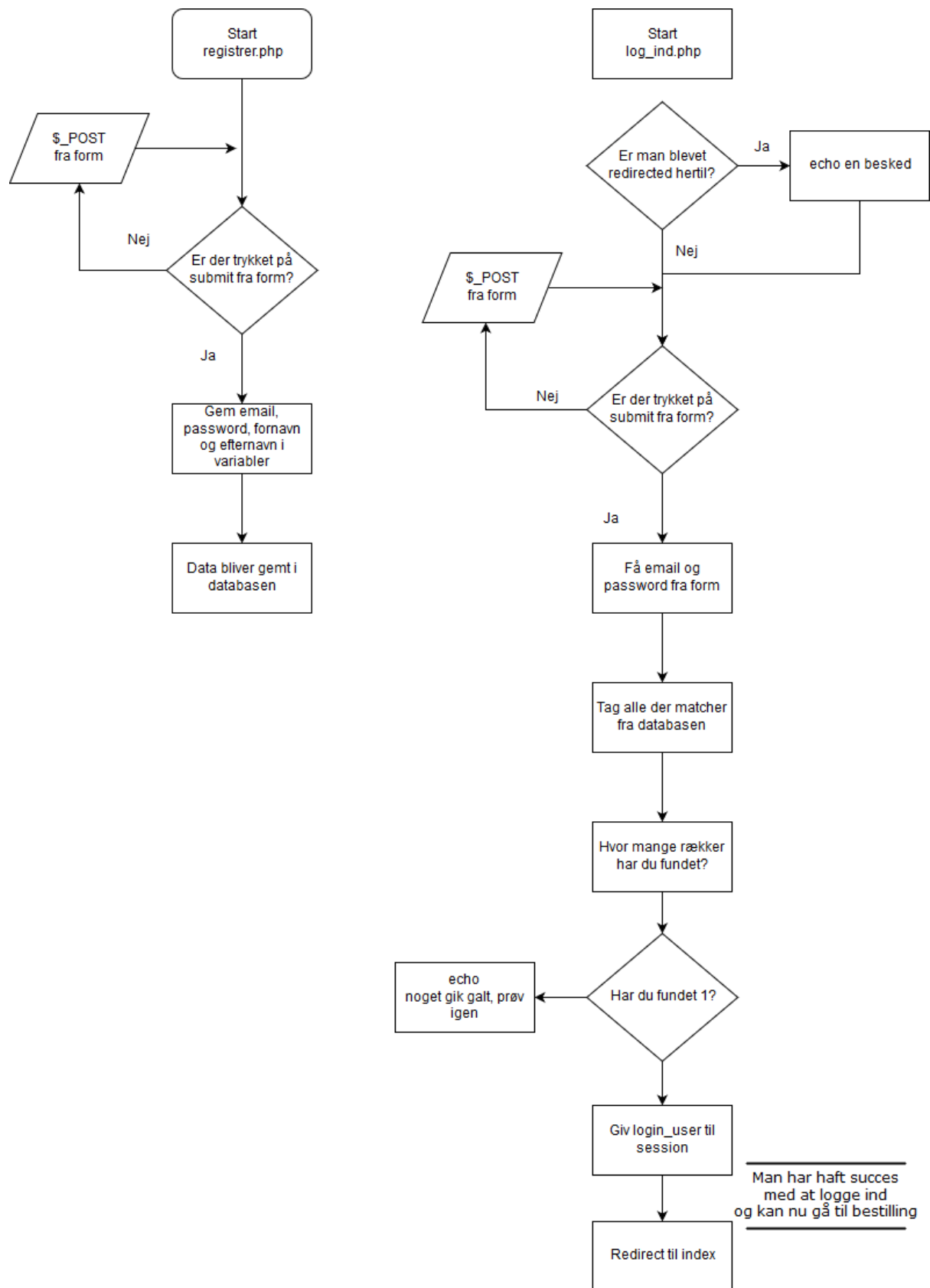
Figur 18 linje 24-29 i bestil.php. Viser hvad der sker hvis man er logget ind

En anden ting der er vigtig at notere, er at vi på alle sider har brugt den samme nav-bar, men alt efter hvilken side man er på, har vi ændret hvilken tag der har classen *active* for at indikere hvilken side man er på.

```
29 <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
30     <div class="container">
31         <a class="navbar-brand" style="color: white;">Bager Bestillingen</a>
32         
33         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#
34             <span class="navbar-toggler-icon"></span>
35         </button>
36         <div class="collapse navbar-collapse" id="navbarResponsive">
37             <ul class="navbar-nav ml-auto">
38                 <li class="nav-item active">
39                     <a class="nav-link" href="index.php">Hjem
40                     <span class="sr-only">(current)</span>
41                     </a>
42                 </li>
43                 <li class="nav-item">
44                     <a class="nav-link" href="log_ind.php">Log ind</a>
45                 </li>
46                 <li class="nav-item">
47                     <a class="nav-link" href="registrer.php">Registrer</a>
48                 </li>
49                 <li class="nav-item">
50                     <a class="nav-link" href="bestil.php">Bagere i nærheden</a>
51                 </li>
52             </ul>
53         </div>
54     </div>
55 </nav>
```

Figur 19 Vores navigationsbar. Som man kan se på linje 38-39. er det taget fra index.php da det er det tag der er sat til *active*

FLOWCHART OVER LOGIN SYSTEM



Figur 20 Dette er et flowchart som viser logikken bag vores login system. Til venstre er registreringen af en bruger, og til højre er log ind af en bruger

TEST AF WEBSITE

Vi har brugt test af website til at finde ud af eventuel videreudvikling til vores website da et kritisk øje kan være meget godt til at sætte ens website i perspektiv. Der er altid noget der kan forbedres og det er, efter vores mening derfor meget vigtigt at man tester sit website og får så meget feedback som muligt.

Vi har brugt Christoffer og Emil som testdukker for at se om vores website rent faktisk fungerer og er brugervenlig overfor folk som ikke har siddet og arbejdet med den. De passer også ind i vores målgruppe som 18-årige dog handler de ikke så meget ind endnu da de stadig bor hjemme, men idéen er også at de skal kunne overskue vores side og synes at den er godt design. Vi startede med at smide dem direkte ind på vores index.html altså vores "hjem" og sagde at de skulle prøve at komme ind og finde deres nærmeste bager. Dette vil vi selvfølgelig have dem til da det var blevet til en af de største grunde til at websitet blev lavet. Det første de gjorde var som forventet at gå op og tryk på "nærmeste bagere" i vores navigationsbar.

Dette videresendte dem til vores login system som de godt kunne regne ud krævede en registrering først. De kunne derfor trykke på vores knap "registrer" knap lige nedenunder log ind knappen og blive videresendt til registrering. Derefter oprettede de sig selv ved at smide informationer ind i vores database og derefter kunne de logge ind hvor de blev videresendt til vores hjem igen.

Dette fik vi feedback på for de godt kunne viderestille til den side man var på vej ind på til at starte med. Det vil sige at siden de var på vej ind for at finde den nærmeste bager vil de gerne blive videresendt derhen i stedet for at komme ud på vores hjem igen.

Efter de blev sendt ud på index.html igen var det ret åbenlyst for dem at de bare skulle gå ind på bagere i nærheden igen som til at starte med og derefter kunne de også nemt finde ud af at vælge vejle i select boksen.

Select boksen sagde de også gerne at de vil have lavet lettere så Google Maps tog din lokalisering i stedet for at du selv skulle vælge din position i landet og det havde vi også tænkt over at gøre selv men det var både besværligt at kode ind og så tænkte vi at det også vil være smart at lave det andet system hvis man nu ikke skulle bestille brødet til sig selv. Ved vores system vil du fra fx Vejle kunne bestille brød eller kage til Give hvis man nu skulle overnatte der dagen efter. Vi valgte så på grund af de to ting at det ville være bedre at du selv kunne vælge hvor du ville bestille det fra.

Udover de to ting synes de at det var et helt fint website med godt og overskueligt design og rimelig nemt at gå til.

KONKLUSION

I vores projektbeskrivelse fandt vi den problemstilling at der ikke var nogle gode websites til at bestille fx kager og brød online altså deres layout var enten ikke ret godt eller også var det uoverskueligt at navigere rundt i. Vi har så lavet en website som skal fikse dette problem. Vi har samtidigt også skulle tænke på at vi skulle have overvågning til at indgå i vores website. Den opgave fik vi jo så klaret bl.a. ved at lave et registrer- og login system ved hjælp af en database, hvor vi gemmer brugerens fornavn, efternavn, e-mail og adgangskode. Den vej kunne vi opnå nogle informationer fra dem.

Det vi havde tænkt med siden til at starte med var jo at man skulle kunne bestille brød eller kager fra vores website men vi er endt ud med at lave en website hvor man ved hjælp af Google Maps kan finde sin nærmeste bager og besøge deres hjemmeside. Hvis vi kunne videreudvikle websitet, vil vi klart have haft en bestillings funktion for lige nu er det i bund og grund bare en hub til at navigere rundt i en masse bagerforeninger. Man kunne i princippet bare gå ind på Google Maps og søge på bagere og så svarer det til halvdelen af det vi har på vores website.

Vi kan også ud fra den feedback vi fik fra test af website konkludere at det ikke altid er det bedste valg man træffer da der altid vil være andre der kan komme op med et andet design eller en bedre idé til at gøre noget man selv vil have lavet på en helt anden måde. Fx havde vi ikke lagt mærke til at man måske egentlig godt kunne videresende brugeren til den side som de var på vej ind på før de loggede ind. Vi havde bare tænkt at det var fint så længe vores login system kunne fungere som det skulle og kunne låse de ting op som krævede et login.

EVALUERING

Vores projekt omkring overvågning er generelt gået rigtig godt synes vi selv. Vores arbejdsfordeling har været rimelig lige fordelt. Vi har brugt hinandens styrker til at få arbejdet til at forløbe lettest for den ene er fx ikke nær så god til PHP som den anden og den anden er bedre til tekst skrivning osv. Så vi har ikke ligefrem trukket hinanden ned som vi hver især, har oplevet i andre projekter.

Fremadrettet har vi besluttet os for at vi gerne vil være færdig med vores "produkt", altså i det her tilfælde vores kode, i god tid. Vi har ikke fulgt tidsplanen særlig godt pga. manglende moduler i en uge så vi kom en smule bagud da det var der hvor vi skulle have lavet vores kode. Det vil sige at næste gang skal sørge for at lave noget derhjemme og ikke kun være afhængige af stabile moduler hver uge. Vi kan dog stadig blive enige om at vi ikke har været stresset på nogen måde over ikke at være helt færdige med det hele løbet af de sidste par dage.

REFERENCER

Bootstrap. (u.d.). *Bootstrap*. Hentet fra Documentation: <https://getbootstrap.com/docs/4.1>

Brød og Kager ApS. (2018). *Kringlexpressen*. Hentet fra Kringlexpressen: <https://kringlexpressen.dk/>

Goldmaker, T. L. (25. 04 2018). *TheLazyGoldmaker*. Hentet fra Forside: <https://thelazygoldmaker.com/>

Google Inc. (u.d.). *Google Maps API*. Hentet fra Maps Embed API:
<https://developers.google.com/maps/documentation/embed/guide>