



## Gæstebog med validering opbygget med MySQL

**Dette er en simpel gæstebog, som kan hjælpe folk med at lave en velfungerende gæstebog uden alt for meget arbejde. Gæstebogen benytter MySQL til at gemme posterne og alle input-felter bliver valideret inden de bliver indsat i databasen. Alle dele er grun**

Skrevet den **06. Feb 2009** af **mstorgaard** | kategorien **Programmering / PHP** | ★★★★★

### Indledning

Jeg har her lavet en simpel gæstebog, som sørger for at validere brugernes input. Jeg har med vilje undladt brugersystem, da jeg ikke finder det nødvendigt på en gæstebog, og at artiklen derved ville blive for omfattende. Gæstebogen er bygget simpelt op uden for mange overflødige ting.

### Indhold

Gæstebogen indeholder validering af alle felter, da det skal sættes ind i en database. Der tjekkes derudover om navn og e-mail er korrekt.

### Funktionen

```
<?php
$db = mysql_connect("localhost", "root", "kodeord");
mysql_select_db("database", $db);
$result = mysql_query("SELECT * FROM gaestebog ORDER BY dato DESC;");
while($row = mysql_fetch_array($result)) {
echo "<b ><a href=\"\" . $row['email'] . \"\">\" . $row['navn'] . \"</a>[/b] (\" . date("d/m, H:i", $row['dato'])
. ") : \" . $row['besked'] . \"<br />\";
}
if(isset($_POST['navn'])) { $navn = $_POST['navn']; }
if(isset($_POST['email'])) { $email = $_POST['email']; }
if(isset($_POST['besked'])) { $besked = $_POST['besked']; }
if(isset($_POST['submit'])) {
if(empty($_POST['navn']) || empty($_POST['email']) || empty($_POST['besked'])) {
if(empty($_POST['navn'])) { echo "Navnet er ikke udfyldt<br />"; }
if(empty($_POST['email'])) { echo "E-mailadressen er ikke udfyldt<br />"; }
if(empty($_POST['besked'])) { echo "Beskeden er ikke udfyldt<br />"; }
} else {
if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn']) || !preg_match("^([a-zA-ZæøåÆØÅ0-9_-]+@[a-zA-ZæøåÆØÅ0-9_-]+\.[a-zA-ZæøåÆØÅ.]{2,5})$", $_POST['email'])) {
if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn'])) { echo "Navnet er ikke korrekt<br />"; }
if(!preg_match("^([a-zA-ZæøåÆØÅ0-9_-]+@[a-zA-ZæøåÆØÅ0-9_-]+\.[a-zA-ZæøåÆØÅ.]{2,5})$",
$_POST['email'])) { echo "E-mailadressen er ikke korrekt<br />"; }
} else {
$besked = mysql_escape_string($_POST['besked']);
$time = time();
mysql_query("INSERT INTO gaestebog VALUES(' , '{$_POST['navn']}', '{$_POST['email']}', '$besked',
'$time');");
echo "Beskeden er sendt<br />";
$navn = "";
$email = "";
$besked = "";
}
```

```

}
}
}
?>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
Navn: <input type="text" name="navn" value="<?php echo $navn; ?>" /><br />
E-mail: <input type="text" name="email" value="<?php echo $email; ?>" /><br />
Besked: <textarea name="besked"><?php echo $besked; ?></textarea><br />
<input type="submit" name="submit" value="Send" />
</form>

```

Det der skal sættes ind i databasen er følgende udtryk. Der indsætter jeg hver af de dele der skal bruges, hvilket vil sige id, navn, e-mail, besked og dato. Id bruges til at give hver besked et unikt navn i databasen, navn bruges til at fortælle hvem afsenderen er, e-mail bruges som kontaktinformation på afsenderen, besked bruges til vise hvad afsenderen har skrevet og dato bruges til at fortælle, hvornår beskeden er sendt:

```

CREATE TABLE `gaestebog` (
  `id` INT( 16 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `navn` VARCHAR( 255 ) NOT NULL ,
  `email` VARCHAR( 255 ) NOT NULL ,
  `besked` TEXT NOT NULL ,
  `dato` INT( 10 ) NOT NULL
) ENGINE = MYISAM ;

```

Denne del bruges til at forbinde til databasen. Første del er til at forbinde til selve databasen, hvor man indtaster henholdsvis host, bruger og kodeord. Herefter vælger vi database:

```

$db = mysql_connect("host", "bruger", "kodeord");
mysql_select_db("database", $db);

```

Denne del trækker data ud af database. Første linje finder al data i tabellen "gaestebog", og sorterer dem efter, hvornår de er oprettet i faldende tilstand.

Herefter bruger vi en "while"-løkke til at trække data ud, og skrive dem ud, hvilket vil komme til at se sådan her ud: "Navn (01/01, 00:00): Besked". Til sidst lukker vi løkken:

```

$result = mysql_query("SELECT * FROM gaestebog ORDER BY dato DESC;");
while($row = mysql_fetch_array($result)) {
echo "<b ><a href='\" . $row['email'] . \">\" . $row['navn'] . \"</a>[/b] (\" . date(\"d/m, H:i\", $row['dato'])
. \"): \" . $row['besked'] . \"<br />\";
}

```

Denne del gør at variabler bliver tildelt felternes indhold. Dette bruges til at udfylde felterne igen, hvis der fremkommer en fejlmeddelelse:

```

if(isset($_POST['navn'])) { $navn = $_POST['navn']; }
if(isset($_POST['email'])) { $email = $_POST['email']; }
if(isset($_POST['besked'])) { $besked = $_POST['besked']; }

```

Denne del sætter processen i gang, hvis der er blevet trykket på Send-knappen:

```
if(isset($_POST['submit'])) {
```

Denne del tjekker om alle felterne er udfyldt. Først tjekker den om nogen af felterne er tomme, og hvis en af dem er tomme, går den videre og tjekker hvilke eller hvilket felt, som ikke er udfyldt. Til sidst sender den os videre, hvis felterne er udfyldte:

```
if(empty($_POST['navn']) || empty($_POST['email']) || empty($_POST['besked'])) {  
if(empty($_POST['navn'])) { echo "Navnet er ikke udfyldt<br />"; }  
if(empty($_POST['email'])) { echo "E-mailadressen er ikke udfyldt<br />"; }  
if(empty($_POST['besked'])) { echo "Beskeden er ikke udfyldt<br />"; }  
} else {
```

Denne del tjekker om navn og e-mailadresse er udfyldt korrekt. Først tjekker den, via regexp, om nogen af dem indeholder fejl, og hvis de gør, går den videre og tjekker hvilke eller hvilket felt, som indeholder fejl. Til sidst sender den os videre, hvis navn og e-mailadresse er korrekt udfyldte:

```
if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn']) || !preg_match("^([a-zA-ZæøåÆØÅ0-9._-]+@[a-zA-ZæøåÆØÅ0-9-]+\.[a-zA-ZæøåÆØÅ.]{2,5})$", $_POST['email'])) {  
if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn'])) { echo "Navnet er ikke korrekt<br />"; }  
if(!preg_match("^([a-zA-ZæøåÆØÅ0-9._-]+@[a-zA-ZæøåÆØÅ0-9-]+\.[a-zA-ZæøåÆØÅ.]{2,5})$",  
$_POST['email'])) { echo "E-mailadressen er ikke korrekt<br />"; }  
} else {
```

Denne del sender beskeden. Først sørger den for at besked-værdien ikke indeholder nogen farlige tegn, hvorefter den definerer variabelen \$time, så man har et timestamp på hvornår beskeden er oprettet. Herefter indsætter den beskeden i tabellen "gaestebog". Til sidst skriver den at den har sendt beskeden, så brugeren også ved det:

```
$besked = mysql_escape_string($_POST['besked']);  
$time = time();  
mysql_query("INSERT INTO gaestebog VALUES('', '{$_POST['navn']}', '{$_POST['email']}', '$besked', '$time');");  
echo "Beskeden er sendt<br />";
```

Denne del bruges til at nulstille variablerne til felterne. Det er for blandt andet at undgå åbenlyst at lægge op til dobbeltpost eller spam:

```
$navn = "";  
$email = "";  
$besked = "";
```

Denne del bruges til at afslutte de forskellige dele. Den er der ikke så meget at sige om, men syntes lige den skulle med (:

```
}  
}  
}
```

Og får lige at få de sidste tabte sjæle med, så kan jeg fortælle at jeg bruger dette til at indlemme mine PHP-koder:

```
<?php  
?>
```

Denne del er selve formen. Først definerer jeg hvilken måde jeg bruger til at sende med, og derefter bruger jeg \$\_SERVER['PHP\_SELF'] til at udføre vores submit i denne fil. De næste er inputfelterne, og så indsætter jeg en værdi, hvis det skal bruges. Dem har jeg defineret i starten af funktionen, og hvis e-mailen er blevet sendt er de variabler så blevet tømt igen. Den sidste del er knappen, som jeg bruger til at sende. Det vigtige ved den er at den hedder submit, da det er den, som sætter funktionen i gang, når der er blevet trykket sendt.

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">  
Navn: <input type="text" name="navn" value="<?php echo $navn; ?>" /><br />  
E-mail: <input type="text" name="email" value="<?php echo $email; ?>" /><br />  
Besked: <textarea name="besked"><?php echo $besked; ?></textarea><br />  
<input type="submit" name="submit" value="Send" />  
</form>
```

## Afslutning

Tak, fordi du læste med, og jeg håber, du kan bruge min artikel til noget.

Giv endelig konstruktive kommentarer.

## Opdateringer

27-02-2007: ÆØÅ-validering tilføjet

27-02-2007: ereg() erstattet med preg\_match()

### Kommentar af wicez (nedlagt brugerprofil) d. 01. Mar 2007 | 1

Meget rodet kode, med mange unødvendige linier - eksempelvis er der ingen grund til at bruge andre variable for \$\_POST-variablene.

### Kommentar af danieru d. 26. Feb 2007 | 2

Enig med windcape, rodet kode :o)

men virker, det gør det da :P

### Kommentar af lylover d. 01. May 2007 | 3

Kunne godt tænke mig du prøvede ikke at undlade bruger system.. Tror man ville lære mere af det ;)

### Kommentar af windcape d. 26. Feb 2007 | 4

rodet kode, benytter gammeldags ereg (preg er bedre), ingen indents (hvilket gør den ulæselig), bruger PHP\_SELF hvilket ikke er brugbart i et miljø hvor action ikke skal reffere til filnavnet, men også en querystring. Derudover ville en mere moderne tilgang nok være at foretrække. Kig f.eks. på <http://www.dragons-lair.org/module/repository/PHP/Guestbook> for et eksempel.... Men godt forsøgt.

### Kommentar af fwskp d. 12. Mar 2007 | 5

Der findes efterhånden mange "alm." gæstebøger.  
Hvad med at udvide den med noget captcha?