ReView: Registration and Viewing webpage

Final Report

Ólafur Hannesson
Supervisor: Nik Whitehead

School of Computing
Faculty of Business and Natural Sciences,
University of Akureyri.

Resubmitted May 2008, in partial fulfilment of
the conditions of the award of the degree BSc.

I herby declare that this interim report is all my own work,
except as indicated in the text:

Signature_____

Date_____/_____/_____

Abstract

We, the human beings are not the only ones who are registered at birth. Horses are registered and given a identification number, but what is different is that while we are later registered with a name, ID number and where we live, the horse are not so lucky as everything is monitored. The information goes into a database called WorldFengur. Some time ago I got a project from a guy named Steinn Másson who asked me to create a database system with an interface to manage mainly the registration for retail shows as well as organization for them. The new system has just about 11 tables and numerous websites to work for. It was decided to use MySQL on the database, HTML and CSS to handle the layout of the website and PHP to be around this all. PHP has the role of getting the things in the correct labels at the correct time. Apache takes then care of allowing the user to see what he is doing and making sure that MySQL, PHP and the HTML are talking together. In this report I deliver my final design and implementation.

Table of contents

Introduction

To use Apache, PHP and MySQL is a monumental task in such a way that it can be used for so many tasks that a good sized webpage only scratches the surface. Many choose though not to employ PHP for the security reasons and that applies to MySQL as well. These packages will do well for small things that allow the user to hold around lots of data with minimal fuss, especially since not everyone has access to the big and rather expensive database driven programs out there.

Where I work is a database driven program (suspicions about it being written in java and the database is MySQL) and that program is very heavy to work with and is really expensive to boot. It is however a program that I would not recommend to anyone. What is necessary for any programs is that they should be as light as possible. PHP and MySQL are very light together but then my experience does count one project, this one. This project is very light and fully worked should do a whole lot more than any project with java or JavaScript does, meaning that it should be lot more easily to work with.

This report does contain a bit of a background of the data involved, description of the tools uses, why this project was chosen and how. It also details some design work as well as describing the implementation. The implementation was a lot different from the original design. The report ends with short description of evaluation. In the appendices are listed the code for the project, given more accurate timeframe and some UML diagrams as well as some sample data.

Background

When the settlers came to Iceland some 11 centuries ago, among the animals were horses. These horses were believed to be of Scandinavian horse race and it is from those horses today's horses decent from. Number of tests have been performed to determine which horses today are the most alike the Icelandic horse and there are horses in Norway that comes the closest, even going so far as saying that ancestor of the Icelandic horse is from those horses.

Picture 1: The Icelandic horse

The Icelandic horse has been described as smallish horse but sturdy and hardy with strong characteristics and character and lively temperament (see Picture 1). Unlike many of their relatives around the world, the Icelandic horse has been raised in natural conditions, i.e. they can move around freely alone or in a herd. Those conditions may have contributed more than most to their strong characteristics. People going abroad may find the horses there easily frightened and temperamental. Mother Nature and human beings are the most danger to horses in Iceland but since there is no natural predator of horses here in Iceland, the horses seek strong attachment to people and are quite affectionate.

The official goal is to breed a healthy, fertile and durable horse while preserving all the varieties of the coat colour found within the Icelandic horse race. The horse is a riding horse and as such has many riding abilities. Every horse is unique in its gaits but they all should be able to trot, gallop, walk, pace and slow trot. These styles are in addition to spirit, slow canter and general impression. These abilities are judged in competitions like "Landsmót" (held every two years and is the main focus) with lot of other shows, including but not limited to retail shows and county fair. The judges measure the horse, for example they measure height at the highest point of the withers, height at the lowest point of the back, width of the chest and hooves. The judges give the horses marks based on their abilities and how the horse is conformed. The assessment system is made up of the goals for breeding in Iceland. [(Finn, Kringeland, Hansen, & Antonsson, 2002)] All Icelandic horses have unique identification number and many horses have a microchip in them. The identification number consists of 2 letters and 10 digits (example: IS2002286808)

- "IS" is the country code, country of birth for the horse
- "2002" is the year of birth
- "2" is the sex of the horse, "1" being the male horse, "2" being the female.
- "86" being the area the horse is born in
- "808" is the number the owner gives the horse.

As I have said before, the database in use for Icelandic horses today is WorldFengur. WorldFengur is a joint effort by the FAIC (Farmers Association of Iceland) and FEIF (International Federation of Icelandic Horse Association) to collect data and store at one central location about horses of Icelandic origin in the world. In WorldFengur you can find approx. 300 000 Icelandic Horses with information on owners, breeders, offspring's, pedigree and so on [(Ahola & Lorange, 2001)]. The database was created in the year 2000 and is only accessible through the internet. Recently, people have been able to put in and access DNA information for the horses.

My project

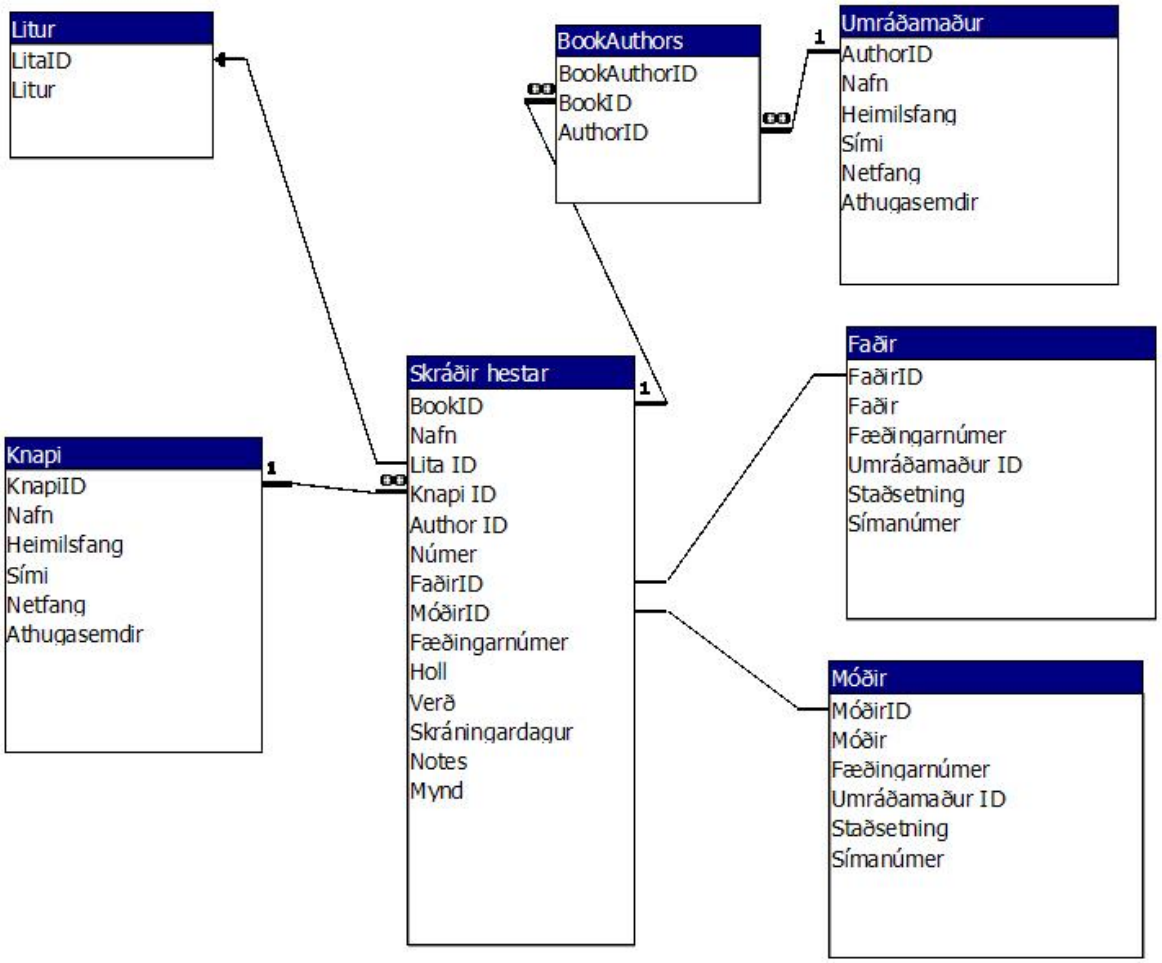
The Project

nr :		**Holl :**	
Nafn :	_	**Verð :**	_
F.nr :	_	**Knapi :**	_
Litur :	_	**Umráðam :**	
F :	_	**Sími :**	_
M :	_	**Netfang :**	_
Lýsing :			_
	_		_

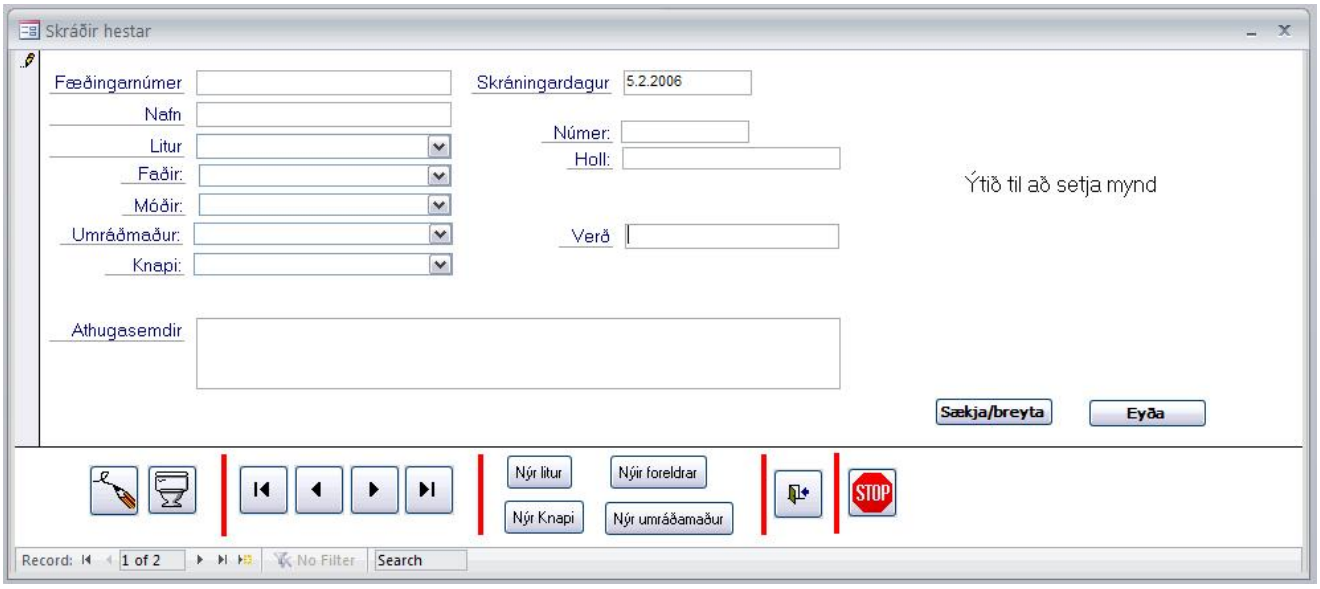
Table 1: Typical registration form

In a way, my project is similar to WorldFengur. WorldFengur holds lot of information but does not hold all. The interface will offer the user to enter much of the same information as WorldFengur does but will also give the user the ability to organize horses such as which horses are going to be on the retail show, which horses are not. Today, owner of a horse sends Stein an email with the registration of the horse calls him on the phone or sends it to him in a fax. When he came to me and asked me to create that "something", I started by creating a database in Microsoft Access with an interface. That database was very simple as is shown in Relationships 1. There is one main table called ´Skráðir hestar´ which holds the information about the horse itself. In the table are links to other tables, holding the owner or caretaker ´Umráðamaður´, colour ´Litur´, father of the horse ´Faðir´ and mother ´Móðir´ and finally the jockey ´Knapi´. Table 1 shows how the typical registration form looked like, but this form was filled in after the information from the registration, then printed out and used at the retail show. As it was necessary, they did not need too much information, so the database did not need to be big nor was the interface anything special, see Picture 2.

In this first version the user needs to take all the data and set it in manually as usual, however, if the time allows me, I will put in place a function that allows them given that the file is of the correct format and the information is in correct place in the file, to import the file into the database. The ability to import the file will rest on the look of the registration form (Table 1). Some modifications may be done later on the registration form but in essence, it will be the same and look the same. The user then will be able to type in missing information for the horse later on when he has time to spare.



Relationships 1: How the old database looked like



Picture 2: the old interface in Access

In this project, the user will be able to print out or save down on the desktop to review later, ready-made reports, search the database for the information required and make some changes. When user is entering information into any form, the program is constantly checking if the user has entered similar information before and when the user submits it, there will be performed a check to determine if the user has filed that particular information before. If the user has done that, there will be message which confirms it, otherwise the user will be able to view the data to confirm everything is correct and then confirm it to send it to the database. In the event the user gets the message that the horse has already been registered, he will have 2 option, to go back and check the data he typed in, again and fix that or discontinue and end on the main page.

The project is made for the individual computer. That means each user will have their own copy of the database with an interface. Given time and more people showing interest into ReView, I will create central webpage and allow the user to connect to this central webpage to get information or send. That webpage can then be a central place where people could keep track of retail shows, see who is selling which horse and which horse is doing well if people will allow.

My tools

There are some tools out there that are available to create a database driven webpage but they are as many as they are different. I chose Apache and MySQL because I was at least familiar with them but once started researching, decided on PHP because of 2 factors, first, I did not know it but looked very interesting and second I wanted to learn something new.

What is Apache? According to the Apache website, "*The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful and freely-available source code implementation of an HTTP (Web) server.*" [(Behlendorf & al, 2002)] Apache HTTP Server or more commonly Apache is the market leading free web server (according to NetCraft [(NetCraft, 2007)]). While Apache is developed and supported by the open source community, it is freely distributed and is easy to set up. Many website designers have installed and set up Apache on their machines to enable easy way to preview and test websites while being developed. Apache was first created by Robert McCool sometime in the early 1990s with its next version released in 2002. Today's version is 2.2.6, released 7. September 2007. I downloaded Apache version 2.0.59. I did not go with earlier version for the simple reason that there were some compatibility issues between Apache, MySQL and PHP (the newest versions available ate the time). Today, 12. January there are still some issues and it is recommended to use some older versions.

PHP or PHP: Hypertext Preprocessor is a programming language that was designed for producing dynamic web pages. It is still a popular misconception that PHP stands for Personal Home Page Tools as its name was in the beginning. According to the official PHP webpage, PHP is "*a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML … the main goal of the language is to allow web developers to write dynamically generated web pages quickly.*" *[* (PHP: Hypertext Preprocessor, 2007)*]* PHP is easy to use when the intent is to create simpler websites from some very complicated and vast amount of data. Its most popular choice is to use in server side scripting but is also used in graphical applications or from a command line interface. PHP is open source server-side language that is very easy to learn not

to mention very powerful. PHP is becoming very popular choice along with MySQL and Apache to create database driven web pages. The reason for its popularity might be very simple. PHP uses simple syntax, i.e. the scripts are kept as readable as humanly possible which in turn allows for flexibility. This flexibility however leads to security concerns as well as performance issues.

Rasmus Lerdorf is titled the father of PHP when he started with it in 1994. He used if mainly for maintaining his personal home page. PHP2 came out year later and PHP3 in 1997. Today's version is number 5.2.5 and was released 8. November 2007. I downloaded that version and set up but the version I had before was number 5.2.3. I have not yet had any problems with this version and Apache but I fully expect them to occur.

MySQL is a high performance multi-user relational database management system that today is the most popular choice when it comes to database driven web pages. The reason for its popularity is preferably the easy way it has when it comes to maintaining it. MySQL is open source SQL database system that has been developed, supported and distributed by MySQL AB which was founded by the MySQL developers. The current stable version of MySQL is 5.1.22 but the newest version is 6.0.3.alpha, released unknown. I did get 5.0.45 since I had some major problems concerning communications between the webpage and the database. Other problems I encountered were for example problems handling strings and numbers, issues regarding privileges and connection problems from the website into the database. I have used phpMyAdmin occasionally to help with the database creation and maintenance as well as making sure that everything is as it should be. phpMyAdmin is basically a help program that "*can manage a whole MySQL server (needs a super-user) as well as a single database.*" [(Ratschiller & al, 2007)]

Currently most websites that employ HTML employ CSS as well. HTML stands for Hypertext Mark-up Language and enables us to mark the text so it can function as hypertext on the web. Editors like Notepad++ and Microsoft FrontPage, mark the manuscript pages with symbols that tells the web how to print the page. HTML uses its own set of symbols on how the browsers should display the pages. CSS or Cascading Style Sheet was designed to help the designers with the formatting of the web pages, comparable to the styles in programs like Microsoft Word. CSS provides a way for the designer to specify the formatting in one file and have the changes reflected all over the website instead of having to lookup the formatting on many web pages. By using CSS the designer is not only guaranteeing himself a way of making the life of maintaining the website much easier, but it will also make it so that when a new person comes along, that person can quickly set herself into the outlook of the website.

Everything else

Currently I use Notepad++ version 4.6, a free open source program, as well as having access to phpDesigner 2007 professional from MPSoftware. Both programs have their positive parts and both have their negative parts. Probably the most positive about both programs is the highlighting help. When writing a loop for example, they will highlight the corresponding bracket once you have finished writing the code and are not sure if you have closed the loop or not, then you can select the opening bracket and the corresponding one will light up if present. phpDesigner is more suitable for programming in PHP and MySQL within HTML. phpDesigner also allows you to see more what you are doing in terms of coding, i.e. assisting with function calls as well as completing keywords correctly

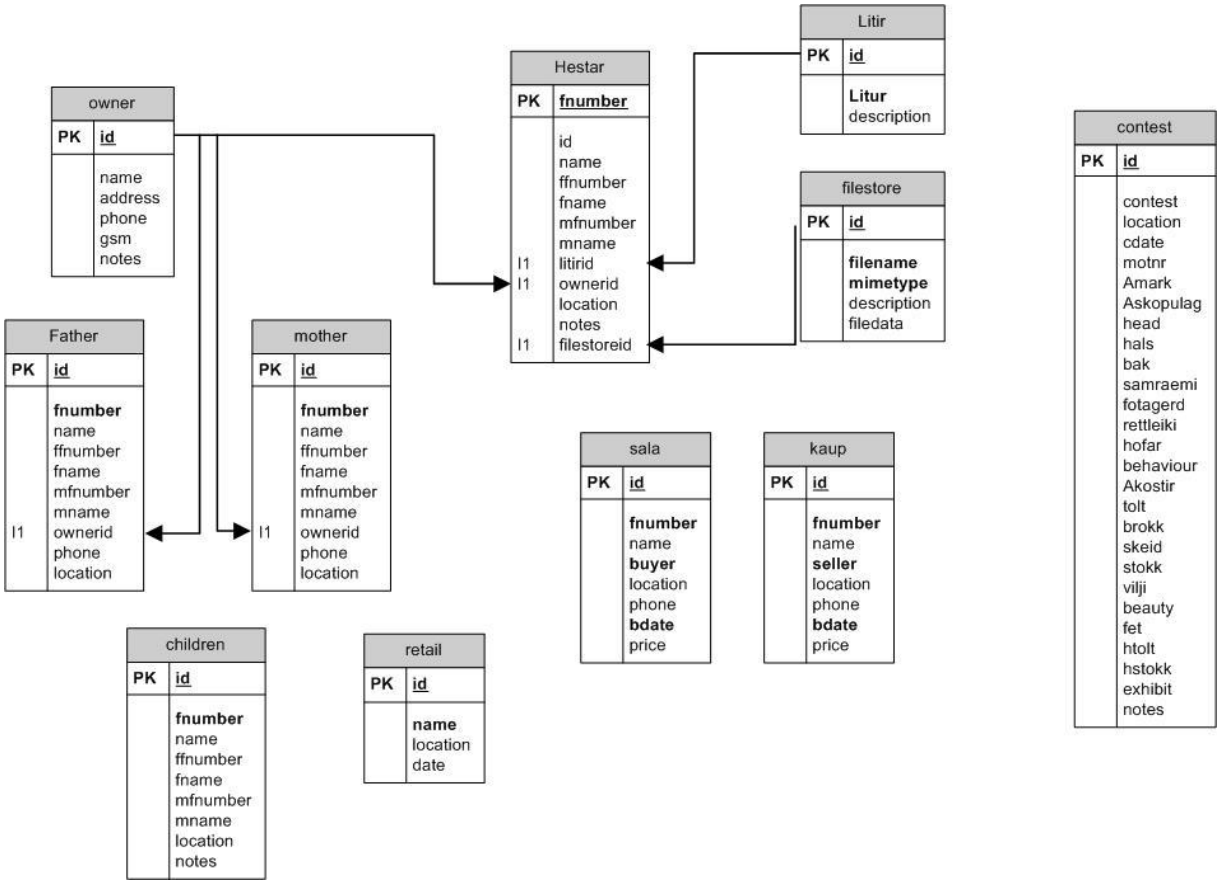
(inbuilt functions). phpDesigner also has more functionality for the programmer while Notepad++ is more of multi-purpose program (useful like a text editor and for coding in other languages).

The negatives about the programs are those that phpDesigner is a bit on the heavy weight in terms of memory and processing. Although it says that in system requirements it needs at least 256 MB RAM or higher, it is probably better of with more than 512 MB RAM. Notepad++ does not take much space, does not need much memory and overall is light. The setting is both small and good and is not enough at the same time. The setting becomes a bit cumbersome at times when working on a file. Of those two programs, phpDesigner is better but still there are things that are better to work with in Notepad++.

I use Mozilla Firefox, version 2.0.0.6 and Internet Explorer version 6.0.29. Personally I use only Mozilla Firefox, but there are number of web sites that are configured respond only to Internet Explorer, so out of necessity I use it. In this project I will however make use of all browsers I can acquire (e.g. Opera, Netscape) onto my computer to make sure that there are no compatibility issues later on.

My solution/Design

Relationship 1 shows how the database looked in the beginning. After some discussions between me and Steinn, it was decided to expand this a bit. To compare the databases, there are some noticeable differences. Picture 3 shows the database as it is today, 13. January 2008 and there



Relationships 2: The initial database layout

is a very strong possibility that there will be some changes at later stages of the creation but without doubt some fine tuning will be made on the interface and database.

What Steinn wanted to see on the interface, was a way for him to register horses for shows, keep a list of all the horses and to be able to make changes. Those changes needed to include a way for him to be able to select a horse from a list of horses and make changes to the data about the horse, to select a number of horses and have those horses appear in a list that he could then print out. The information he wanted to put in the old database was the horse itself, the identity number, name, name of the father, name of the mother, colour, in which group to place the horse, give it a number, registration date, price and a photo. The ability to add some notes was added on the interface later on. I then added another window for Stein to add some more colours, register a new owner, and address the ancestry and the jockey. This is all found more or less in the new interface and database, although under some very different names.

The new database has some 11 table but there is very little relationship between them now but some attributes depend on other attributes to be able to work properly in the registrations. I am not saying that there is no interaction at all just that the information is set in one table instead of having one huge table where the chances of repetition is just too much. Relationship 2 shows the current tables.

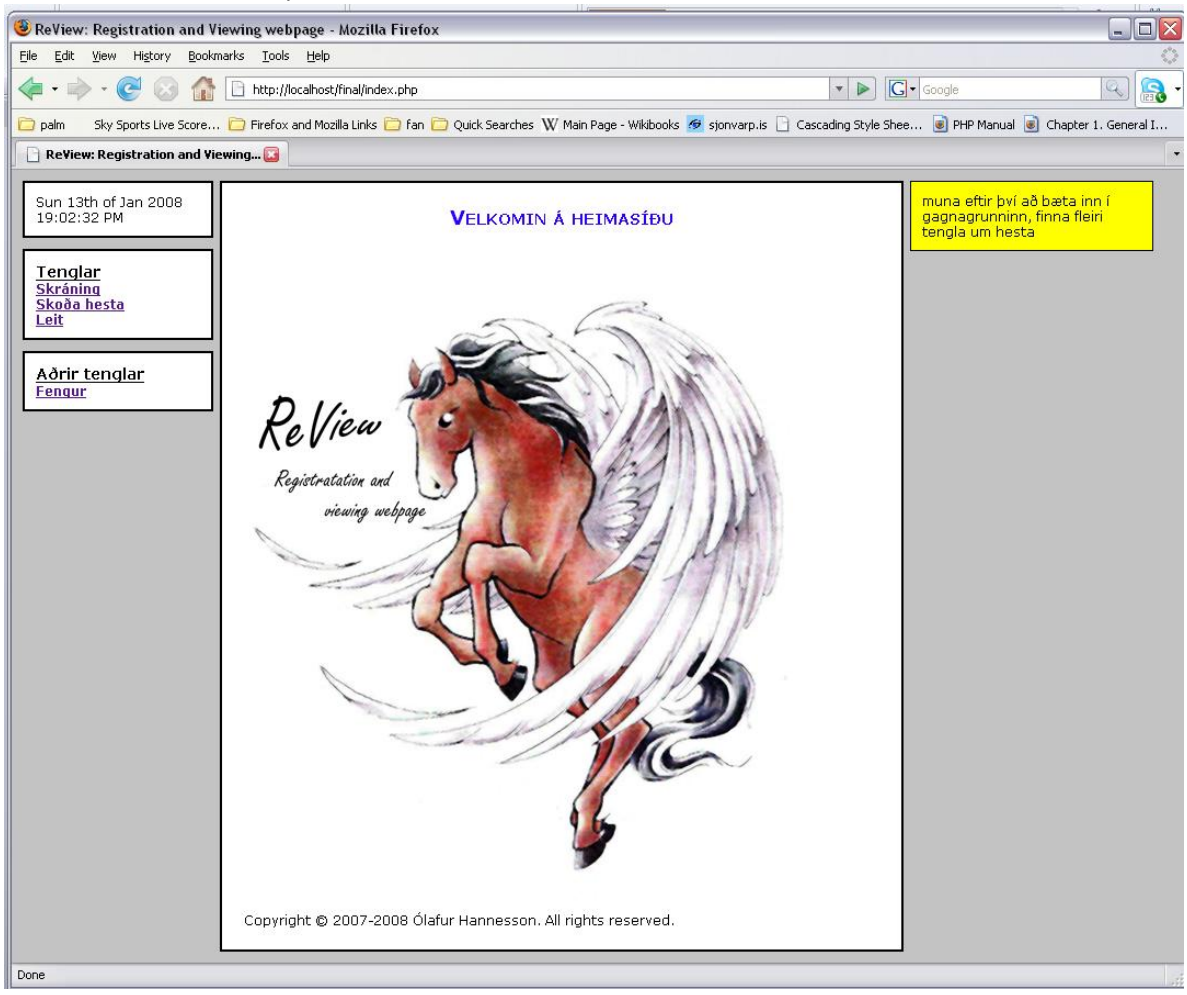
The primary key for all table are defined as *PK*, the user must fill out the corresponding labels in the interface where the bold words are in the database. Those labels define each table. The id is only a auto increment and the primary key for most of the tables but is necessary because when the user uses a drop-down button, checkbox or a radio button, I can show the information registered under that button/checkbox given that I use a select statement to bring it out from the database.

The table *Hestar* is the main table in the database. In the table I store the information about the horse itself. The user files some of the information about the horse himself but the table *Litir* is passed by reference just as information from tables' *owner* and *filestore. fnumber* is the unique identification number for the horse, *name* is just the name of the horse if it has one, *ffnumber, fname* and *mfnumber, mname* are the identification numbers and the names for the horse parents. *litirid* is the id from *litir* table and basically is a reference for the colours, *ownersid* is id from the table *owner.* Location is where the horse is located since the owners often have some one else taking care of the horses for them. *notes* means just that, notes. There the user can set some observations or something like that. *filestore* is passed by reference and is a reference to the pictures for the horse.

The table *filestore* stores the pictures. The *id* is an auto incremented number, *filename* is the name of the picture but here it is more important than ever to have unique names on the files, *description* allows the user to put in a description on the picture. The last two, *mimetype* and *filedata* are bit trickier. Let's start with *mimetype.* MIME stands for *Multipurpose Internet Mail Extensions* and is a code to specify the media type of the file you are submitting. „*MIME defines mechanisms for sending other kinds of information in email, including text in languages other than English using character encodings other than ASCII as well as 8-bit binary content such as files containing images, sounds, movies and computer programs.*" [(Converse, Park, & Morgan, 2004)] This is when the

browsers take the MIME type, e.g. `text/plain`, `image/gif`, `image/jpeg`[1], etc, of the file and check which type it is. When making the table ready for inserting the images, the size of the file becomes a factor because that must be defined. *Filedata* is a binary data. There are several BLOB column types designed to take and store variable length binary data. The ones I was interested in were TINYBLOB, regular BLOB, MEDIUMBLOB and LONGBLOB. These BLOBs take different sized files, TINYBLOB takes 255B or 255 characters, regular BLOB takes 65KB or 65535 characters, MEDIUMBLOB takes 16.8MB or 16777215 characters and the biggest is LONGBLOB that takes 4.3GB or 4294967295 characters[(Yank, 2004)]. I decided to go with MEDIUMBLOB, for the simple reason that I seriously doubt that the pictures will go above that size but are more likely to be in the region of 100KB to 200KB depending on the quality of the picture.

The table *owner* holds the information about the owner, name, address and contact information, the *retail* table holds the information about the retail shows, i.e. if it has a name, location and date. *Litir* holds information about the unique coat colours the Icelandic horse has, the colours name and in some cases a description. In a window on the interface, the user can put in a information about horses he has either sold to another or bought somewhere and the data goes into the tables *sala* and *kaup*. The tables are almost identical and could be set into one table. This is one

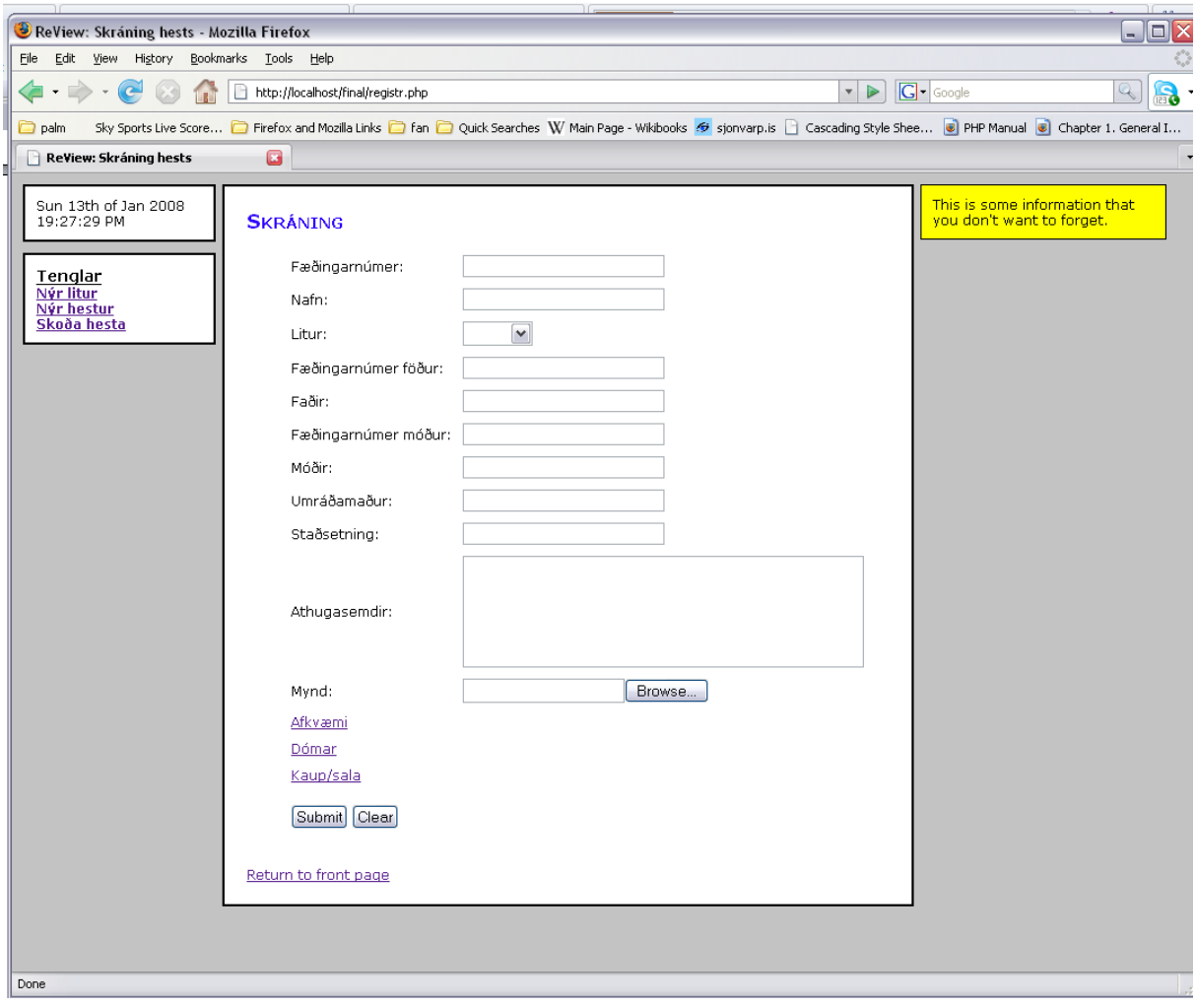


Picture 3: The initial interface

[1] Browsers do not necessarily use the same MIME type for files. For example, uses Mozilla image/jpeg for jpeg images while Explorer uses image/pjpeg.

of the cases of how I may change things as the project goes on. The same applies to tables *Father and mother*. These tables are almost the same and can be set into one table. The biggest table is the *contest* table. This table stores all the data about assessments in the contests the horse has gone to as well as putting the name, location and date of the contest. The user can also put in the name of the exhibit and the number the horse had in the contest.

The interface was perhaps the most difficult to decide. Do we go all out in our constructions or do we employ KISS (keep it simple stupid!)? For this project it was decided to keep it simple, reasoning that there will be always time to go back later and design the outlook properly. So, the first thing to do was to create some basic layout to have on hand to apply to all web pages I would need to create. Picture 3 shows how the main page looks like. The yellow box to the right of the picture will be used to put in tips to help the users around when he is working in the interface. On the front page is as well links to the registration window in the boxes at the left hand side, if the user wishes to look at the list of horses, the user has put in without setting any parameters and if the user wishes to search for something specific. The top box will include the date (e.g. Sun 13th of Jan 2008 and current time) and will be showed on every page. The last box on the left hand side will hold some links (some that will already be configured there and possibly for the user to put in). The search window will only hold some basic labels that the user will have to fill in, should he want to search for anything specific. Failing finding anything, there will be a label the user will fill in and the database will then be



Picture 4: The initial registration form (incomplete)

searched matching anything in the database with the word(s) the user put in. The results for the search will be displayed on the same page below the search form.

The view form will show list of the horses with identity number, name and owner displayed in four-column row (the first would be the id). The identity number would then act as a hyperlink which would then be clicked and the user can then view all available data about he horse, including in separate windows, the offspring's, assessments and owners history if available. Should the user wish to change any data, there is then a button at the bottom offering the user to click and open the registration form in which the user will change relevant data accordingly and save changes to the database. From the registration window the user can fill in also in which retail shows the horse has already been to and when. The user can also control more about the colours, delete colour and add colour and add description to the colours. The user can also make life easier for him and fill out the personnel who work around the horses, included and not necessarily limited owners, caretakers, jockeys, veterinarians, judges and hosts of shows and contests.

The offspring's are entered just as normal horse would be. As is shown in picture 3, the data needed is its unique identification number, name, identification numbers and names for its father or mother depending on whether the horse is a mare or stallion, and the location, since the foal might be in the ownership of someone else.

From the Viewing page the user will be able to go and see some available reports, already configured for him but he will also be able to put forth some criteria to create some reports. Readymade reports will be reports on a horse, the user wants to print out or maybe to send out for someone to look over. Overview over all the horses found in the database, with name, identification number, owner, contact information and location. When the user was viewing the list of all the horses, he ticked a checkbox right next to the name of the horse; he will be able to print out a conclusive list of those horses. These three reports are perhaps the most useful to have but there may be others added. The reports will be done in either plain text format or in a PDF document depending on which the user wants.

Time permitting; I will put in place a function that allows the user to import files of certain formats, e.g. plain text files, word documents, excel files and outlook files, and fulfil certain criteria of how the data in them is set up. If we go back to Table 1 on page 7, that would be preferred outline for the function to get the data from. If not in that setup, then there may be a way to implement a way to use MySQL to take each string and compare them to an example that exists in the tables. However that way, it will take much more time to import each file and depending on how many registrations are in each file, will be very heavy for the project to do, also because there might exist some data in the database, the comparison must be to check the import data against the data already in the database.

Among the last things to do with this project is to have this transformed into a "website in a box". That means that the web interface, the database and Apache will need to be installed onto the computer before the user can start using it. I have researched it a bit and have an idea of what to do and how to do it. My idea is to either have this packed into a .zip file, put on a CD. Then when the disk is inserted into the computer drive, there will be auto run file set to run upon induction, which will then proceed to create a map on the main drive for Apache, PHP and MySQL and put a shortcut for the interface on the desktop of the computer. The second option is the same way, only to have the whole package unpacked on the CD and copied onto the C drive.

Implementation

The tools

I started this project in the beginning with the idea of using PHP, MySQL and Apache. I have not strayed away from that idea and in fact added to that. Html and CSS play big part in the development of this project and there is even a few lines of JavaScript there. Beforehand the only programming languages I had learned about before were html and a bit of MySQL. The biggest challenges in the beginning were to get PHP, MySQL and Apache to gel. Once awhile, when I was experimenting with the languages, error messages flooded in and 3 times I had to start from scratch by uninstalling or removing the PHP and MySQL from the computer and install again, most recently in the beginning of march.

The first problems I encountered were which versions to use. I thought in the beginning that it would all be compatible and downloaded the newest stable versions of PHP, MySQL and Apache. From that time was about month were I battled myself through numerous errors, most of which I could never figure out. I sat down, read more about this and got files that were known to be stable and could work together. The version of PHP I used was 5.2.4, the version of MySQL 5.0.45, and the version of Apache I used is 2.0.59.

In my opinion are very few programs that allow for easy programming in PHP. What I mean, is that unless you know the language inside out to program in Notepad or any such simple editors. There were quite a few that pointed me towards phpDesigner and said that this was the editor to use for anyone creating webpage with database and such as the program allowed for programming in PHP, HTML, CSS and many more languages. phpMyAdmin helped tremendously in latter stages of my project when I needed an access to the database quickly.

When the project of installing PHP, Apache and MySQL on a CD or a USB stick to use failed I searched the internet for solution and found two, while not perfect, it was something I could use. WOS (Webserver On Stick) and XAMPP were the closest packages I could find and could aid me with this project. Both of them are free and have the newest compatible packages available in amongst many other side packages. I went with WOS for the reason that it is simpler to use than XAMPP. When the user plugs the stick in with WOS it is a simple matter of double clicking on the icon to start the packages. XAMPP is simple enough but the user has to manually start each package.

The project

I suffered a bit of indecisive people and sometimes indifference on how the database and the web page should work. There was no shortage of ideas for the web page and only the timeframe that was in the way for me to implement them.

The outlook of the webpage was perhaps the easiest part to do of the entire project. The difference between an outlook and the interface has different meanings in this context. The outlook is concerned with colours, fonts etz., while the interface is concerned with where things belong on the webpage. It was decided upon simple outlook because there is always the possibility later to go all out and create some fancy looking pages. First of all, I decided in the beginning to split the pages up. All necessary files would need to include the main file (see in Appendix main.php) and footer (see in Appendix footer.php). ‘main.php’ holds the beginning of the <html> tag along with the <body> tag and the code for the outlook, the CSS part. Small part is a JavaScript that control the menu in such a way that when the user clicks on one part of the menu, it drops downs, opens if you will, to allow the

user to choose which file he intends to go to. The ´footer.php holds the </body> tag and the </html> tag. Because all the other files include those two files (main.php and footer.php) and the files have the ending .php means that I can have most of relevant tags in two files and don´t need to put them into other files. This also means that any change of outlook of the pages and menu will automatically transfer over to other files. CSS could be set in a separate file but I felt that it was not necessary because this is not that complicated code and in reality very simple. Most of the code is dealing with the outlook of the menu and fonts.

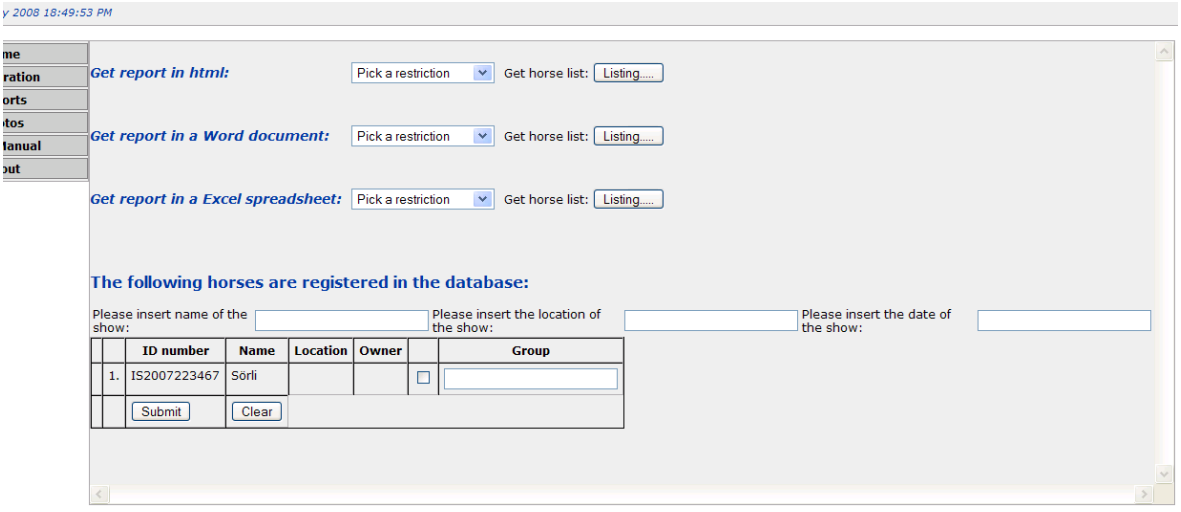
The registration was not simple. There was lot of discussion about how the form was supposed to be constructed and what should be in there. It was decided that the most important information was the ID number of the horse and name, ID number and name of the parents for the horse, as well as owner and location. The option to put in notes was also set there as well as the price. This information is sent to one table with the information for the colour, the parents and personnel pulled from other tables respectively. Once a horse has been registered and sent to a table the user can choose to open in a new window called, 'Take a look' which holds list of all the horses in database table and there the user can view all the information about the horse, or seen some very basic information about it or even delete the horse from the database. The process of deleting a horse is simple. When the user clicks the 'Delete' option in the table, a confirmation window is opened. If the user confirms that he wants to delete the horse a variable is sent, one called "action" which defines what the user is about to do and sends it to the correct place in the code. Then the ID number is pulled from a variable as well and a query is sent to delete that particular horse with that particular ID number. If the query is successful, a message is printed out stating that the horse has been deleted and the user can click on a link to go back or go to the menu to go somewhere else. It was the intention to put in a similar way for the user to edit any horse, i.e. change any information about that horse but there is some problem with the update query. The update query doesn´t seem to work and it might be that I have tried too long and too hard to fix it that I cannot see what the problem is but I am convinced that the problem is a minor one and will definitely be fixed in next version.

One of the problems I faced on the ´Take a look´ page was to get the delete and edit to work as well the ability to view the data about the horses. I fixed that by creating a variable called *action* and put it in a loop. If *action* is empty the code should do nothing but if the variable is set to something put it into the variable *action* and then does something with it. If the variable is equal to view then the user wants to view the information about that horse given that the ID number is set. Should the variable be equal to del, the user wants to delete the horse. Other ideas that I did try but did not work as well as I thought were to use buttons which would when coupled with radio button or checkboxes would delete the horse(s), give the user a chance to edit the data for the horse(s) or view the data about the horse(s). I did try to create separate pages for these things but I felt that would be waste of space and in the long term be a problem (allow for confusion).

The files for the Colour and Personnel, is a very simple forms. The form for *Colour* is simply for the user to type in the name of the colour and if he wants to put a description for the colour. At the bottom of the page is a list of the colours with the description and the option to delete the colour from the database. The *Personnel* form allows the user to put the name, address, occupation and contact information in the form of email, regular phone or gsm phone. The same applies to the *Personnel* form that there is a list, listing all the personnel with all the information and the ability to delete the person.

The *Linage* form is intended for the registration of the potential parents of the horses. Note that any stallion can be father to many horses and any mare can be mother to many. To insure that there will be no confusion about the registration, the user will have to define the gender of the horse, as well as type in the ID number which is what the registration is based upon. The user can also fill in the name of the horse as well as the colour name and ID *number* of father and mother, owner from a drop down list, provided that the person has been registered before, location and contact details. If the gender is male the horse goes into the table that lists males at the bottom of the page or if the gender is female, the horse goes into a table that lists females.

The *Search* form consists of a drop down list giving some fields and an empty text box. The user must be mindful of what he types in, because by typing in “Björk” will not give the same results as “Björk from Hofi”. If the field is chosen correctly the first search term will give results (assuming that there is a horse named Björk in the database) but the latter will not give any results because the string is in 3 parts. It was decided upon this to prevent any lengthy strings in the name field when the registration is done. Should the user decide to put in a string like “Björk from Hofi” he will then know that the horse is in the database will be able to go through the list in ´Take a look´ very quickly. I use the *LIKE* operator in the query and for now put the limit from 0 to 250 horses. Then the query is sent and gets the results back and starts printing out the results. However if the id for the row is empty it will return the message that there is no record found and because there is a new window opened asks the user to close this window if he wants to search again. The problem I faced here, was how to find the string I was looking for. After many tries on using the equal sign in the query I started to explore other options, for example I tried to use the ´MATCH ... AGAINST...’ operators in the query and while that did work, it only worked up to a point. Finally I tried the LIKE operator which is unfortunately case sensitive I started to get the results I wanted. However the user must be mindful about how he types in the information in the database because of the case sensitivity. The results are shown in a table on a page called ´results.php’.

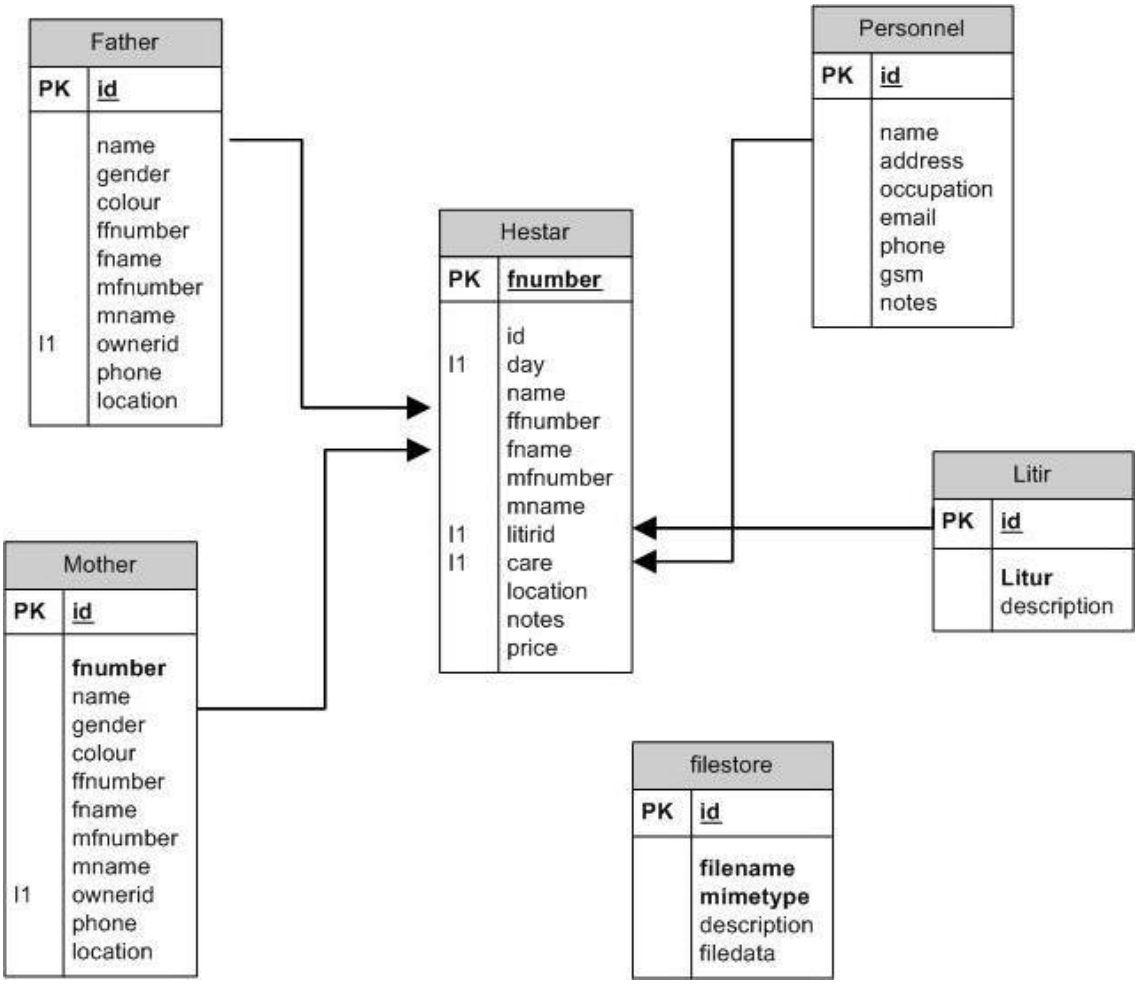


Picture 5: The report form

In the *Reports* page, the user can get reports in html, as a word document and as excel spreadsheet. The user can pick a restriction, meaning that if the user clicks on the drop down button, he can pick under which restriction the horses are sorted by. In Picture 5, here above you can see a checkbox and a textbox. The user can mark the checkbox, set the group and above that table, the user can set in the location of a show, date and name if he wants. The problem is however that the horse will not be set in correct order in accordance with the group the user sets.

The *Photo* upload is a complicated thing. There must be a chance to upload a file, check if it is really a file and if it is print out message that the upload was successful. There must be a way for the page to check which type of image the file is, define the maximum size of the image. The image upload was an afterthought for my parts. The user can upload an image and type in some description of the image and submit it to the database. I managed to get the code to work for any file but could not limit it only to images.

About and *Help* are small parts. About notes the tools used and very short description about how the project came to be. Help is the user manual. It gives a short description about the tools used and gives the user an account step by step on how to install the PHP, MySQL and Apache packages.



Relationships 3: the new database layout

The database consists of 6 tables, table for the colour, any personnel, father, mother, a table for the images and finally for the horse itself. The database layout can be seen in *Relationship 3.* The difference from the old drawing is obvious. 4 tables have been dropped on the request of Steinn for the reason that it is not necessary at this stage to have it in, they might be added later. While I was not told any specifics, there will possible be more data, different data to put in but that is still undecided and unknown.

Problems faced

Some of the problems I have faced with this project have been very strange, not to say the least. However most of the problems around the project were centred on the PHP package with the extensions. There were coming some strange errors telling me that such and such file did not exist when they were amongst the extension files. The error giving me the most headache was 'mbstring.dll does not exist'. This popped up at the most unlikely time and wasn´t resolved until I simply deleted the file named mbstring.dll and put semicolon in front of the extension in the php.ini file.

The biggest problem I faced in relation to this project was when I installed the packages for Apache, PHP and MySQL. I started by getting the newest packages available at the time but very quickly that the PHP version did not work properly with the MySQL or Apache. It did not matter what I did or how, it just did not work together. Later I went and searched for instructions on how to install these packages safely and get them to work together. I found out from the PHP webpage that the packages were incompatible and as such it was recommended older versions. One thing I found out as well was that it can matter in which order the packages should be installed. I started on Apache, then took PHP and ended with MySQL. There is a simple reason for that. Apache must do some changes to the httpd.config file that relate to PHP, PHP must do some changes to php.ini file that relate to Apache and MySQL.

Problems that I faced in the programming were many but most of them feature missing comma, bracket or misspelled word. The one that took most of my time was when I had a drop down list and to get the variable out. The message I got was always that it was an array. So after having spent some time on it, I went into the PHP manual and looked up arrays. This search did not give me much but I wanted to be able to have the page check if the data sent is an array, and if it is, to loop through the data sent and print out each value or give a message that there is no data. I found after much search a variable handling function that finds whether a variable is an array. I knew that I could use the keyword 'foreach' and set any data sent as a variable and print it. That means that it "loops over the array given by *array_expression*. On each loop, the value of the current element is assigned to *$value* and the internal array pointer is advanced by one (so on the next loop, you'll be looking at the next element)."[(PHP: Hypertext Preprocessor, 2007)]

The problem I had with the image upload implementation was that I did not have enough time to put my absolute attention to it. I would have to find a way to limit the sizes of the images, which types are put in and make sure that there is some difference from the names so that when there is a file that has the same name as file inserted earlier, that there will be no problems with duplication in the database. However, there is no worry about which types will be put in, as they will be either .gif and/or .jpg and/or .jpeg.

There is that problem about order of horses in the report section. If the user has decided that horse number 1 will be in group A, horse number 3 in group B and horse number 10 in group A, then that will be the order of the horses, ABA. This is a matter of taking an input, compare it to other inputs, group them together and print them along with the horse the input was at along with it. This has been given much thought but has not been successfully solved. Ideas that were tried but either did take too much time or simply did not work, were to take the data sent, i.e. the ID number sent by the checkbox and the input the user put in the text box, put it into a table in the database. Then it would be fetched again and sorted by group and the horses printed out in correct order. The problem there is the time it takes. If there are many horses it can take some time.

About that database. The biggest mistake I did in this project was not to create the entire interface first and then spend time to develop and create the database. By creating the database first and then the interface as I did, I put in the concept of confusion because there are tables that should not be in the database with columns that should not be along with some fields that are probably not well thought out. Another problem is that given that if the database is not there my code will not work on to create the database with the tables.

It has been very difficult to get PHP, Apache and MySQL to work from a CD or USB stick. In my trials I have found that while settings might work when installed in computer A the chances of it to work in computer B and/or C are slim at best. It is my opinion that I am on right track with it meaning that the paths in the configuration file for PHP, Apache and MySQL need to be set to *I:* (I being the letter of the drive for the USB stick or CD). As to why the configuration does only work in one computer but not the next, is any ones guess but it is my opinion that the configuration of the computer itself prevents it from working in two computers. Once I starting looking around on the internet to find an alternative solution I found a program named WOS and XAMPP Lite. Both are free packages and hold latest compatible packages of Apache, PHP and MySQL. WOS however has some compatibly problems with Windows Vista. Other problems are for example that is will not work if the computer is connected to the internet because then the Apache server will not start. If the packages (Apache, PHP and MySQL) are already installed on the computer, WOS will not start either. I decided to take my chances with WOS because while there is no service with the package, (i.e. no updates and help, beyond posting questions on forums) it is easy to upgrade the package to include other packages, such as phpMyAdmin and Perl. Further tests will be made in the future on getting only the Apache, PHP and MySQL packages to work from either USB stick or a CD.

The problems faced with when I started trying WOS were not many. The one I encountered I solved accidentally. The problem was that when I wanted to start wos.exe the Apache server did not start. I did not have access to the internet where I was at the time so there was no searching on the internet for information. Once I tried some changes in the httpd.conf file (which did not do anything anyway) I shut down Skype (it is configured to automatically start and try to connect to the internet when I start up my computer) and tried to start up wos.exe, and to my surprise it worked. Skype and Apache were configured to listen to the same port number 80, but that does not work.

Evaluation

The evaluation of the project was not that much. A user (Steinn) was set to the task of registering a horse from ground zero and up. There were no apparent problems although he complained about lack of direction which would be my fault as I did not instruct him properly beforehand. The user did register a horses successfully and but did not get the horses in correct order in the reports when he viewed them in the report section. Another problem he found was that if the user wanted to register colour or personnel or parents, he had to go in relevant sections and register them but could not go back to the registration form he had filled part of or all.

Future updates

In order to correct some mistakes and to help with registration, I will introduce sessions in next version as well as introduce drag-n-drop. Sessions will help the user, for example he has to stop in middle of registration, he could save the work he has done and come back to finish later. Drag-n-drop will allow the user to put in information that will then appear on the registration page in some tables that will then allow the user to create something once and then just find it and drag it to correct place in the registration form.

Other updates will have the inclusion of image gallery, to import files that hold registration of horses and the option of .pdf files. Depending on how many will be using this web page, I may add more features as are shown in my initial design.

Final words

As is with everything you can never finish any design until you're done. Through the year I have found out myself that any plan works only so far and most of the time it is better to have plan B and plan C, just in case! This project is on its own a big one, only it does not look like it. In the start I approached the FAIC but did never get any conclusive answers which disappointed me a bit but in a way I am relived since the people answering my questions and offer were toeing the line on being rude. Why, I don't know and simply put I don't want to know.

As you might have noticed I never spoke in length about any related work and there is one very simple answer to that. I only found one project (website) that resembles my work in some way and that is WorldFengur. WorldFengur is a website that is done in JavaScript and has very extensive database behind (I had not gotten any answer on the actual design or what kind of database is used by the time this report was handed in, although I suspect FEIF to use Oracle database). WorldFengur is a second generation website, i.e. has been taken and given a "face-lift" going from being a simple HTML framework website for Icelanders to being written in JavaScript available for all people with horses of Icelandic origin. All in all, there is no website or program that I have found or heard of that is like the project I am doing, but the very few that I have heard of but not been able to get are for very different things.

Before starting on this project I had never used PHP, and only seen MySQL at work in passing lectures in the database course at the university. Apache I had never worked with either so, it was in all a time of learning. This combination of PHP, MySQL and Apache does seem to be very good and very good for web design. I took an interest in this and have found a way to implement this on a USB disk and take with me, although there are some problems with Apache. I will do my utmost to finish this project and continue learning PHP and MySQL beyond this project.

Reference

Ahola, K., & Lorange, J. B. (2001, July 23). Worldfengur - ready for the world. *Information to the breeding leaders of the FEIF countries* . Europe and North America: FEIF.

Behlendorf, B., & al, e. (2002). *The Apache HTTP Server Project*. Retrieved November 8, 2007, from About the Apache HTTP Server Project: http://news.netcraft.com/archives/web_server_survey.html

Converse, T., Park, J., & Morgan, C. (2004). *PHP5 and MySQL Bible.* Indianapolis: Wiley Publishing Incorporated.

Finn, P., Kringeland, I., Hansen, J. S., & Antonsson, G. (2002). *FEIF Rules for Icelandic Horse Breeding.* INTERNATIONAL FEDERATION OF ICELANDIC HORSE ASSOCIATIONS.

NetCraft. (2007, October 13). *Web server survey.* Retrieved January 5, 2008, from Netcraft: http://news.netcraft.com/archives/web_server_survey.html

Pfaffenberger, B., Schafer, S., White, C., & Karow, B. (2004). *HTML, XHTML, and CSS Bible.* Indianapolis: Wiley Publishing Inc.

photographer, U. *Picture of Icelandic horse - Vima from Lækjabotnum.*

PHP: Hypertext Preprocessor. (2007, November Last updated 25.). Retrieved November 8, 2007, from PHP: Preface - Manual: http://www.php.net/manual/en/preface.php

Ratschiller, T., & al, e. (2007). *phpMyAdmin 2.11.1 Documentation.* http://www.phpmyadmin.net/.

Yank, K. (2004). *Build your own database driven website using PHP and MySQL.* Melbourne: Sitepoint.

Table of Figures

Appendix A: Code listing

The order of files

(1) db.inc.php
(2) main.php
(3) footer.php
(4) index.php
(5) registration.php
(6) hestur.php
(7) newcolor.php
(8) personnel.php
(9) line.php
(10) parent.php
(11) mainReport.php
(12) reports.php
(13) reportsN.php
(14) wordi.php
(15) excel.php
(16) search.php
(17) result.php
(18) check.php
(19) photo.php
(20) photoS.php
(21) about.php

1. db.inc.php

```
<?php

/**
 * @author Ólafur Hannesson
 * @email  olafurhannesson@gmail.com
 * @copyright 2007-2008
 *
 * @comment
 *                      this file holds the script for connecting to the database server 1) and
 *                      to the database ´skráðir hestar´ 2)
 *
 */
/** connection established,
 * ´localhost´ = location of the MySQL database server
 * ´root´= username for the MySQL database server
 * ´admin´= the password for the MySQL database server
 */
$dbcnx = mysql_connect('localhost', 'root', 'admin');

/** 1) if the connection fails to the database server */
if (!$dbcnx) {
   exit('<p>Unable to connect to the ' . 'database server at this time.</p>');
}

/** 2) if the connection is ok, we connect to the database we are looking for */
if (!mysql_select_db('skráðir hestar')) {
   exit('<p>Unable to locate the horse ' . 'database at this time.</p>');
}
/*
if (mysql_query("CREATE DATABASE skráðir hestar", $dbcnx)) {
        exit("database created.");
} else {
        exit("error creating database: " . mysql_error());
}


?>
```

2. main.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/XHTML1/DTD/XHTML1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>ReView: registration and viewing webpage</title>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
<!--
            Author: Ólafur Hannesson
            Date: (final changes and commenting) 05.04.2008
            Project: part of final year project autumn semester 2007, spring semester 2008
            Name of file: main.php
            Connected files: 1st part of the front page
            Function: the file holds all information about the outlook of the files for this project as all files include this one
                                    in the header.
-->


<style type="text/css">
/**
* padding widths in the elements
*/
p { padding:10px }

/**
* the configuration of the topmost element, i.e. size of it, the font size
*/
#head {
   height:85px;
   font-size:24px;
   padding: 5px;
   margin-bottom:15px }

/**
* the configuration for the footer, i.e. location, size and font size
*/
#foot {
   position: fixed;
            top: 95%;
            width: 100%;
            height: 35px;
            font-size: 12px }

/**
* how the text is aligned
*/
.sleft {text-align: left; }
.sright {text-align:  right; }
.scenter {text-align: center; }
.sfull {text-align: justify; }

/**
* configuration of the fonts
*/
h1 {font: bold 18px Lucida Handwriting;     color: #003BA1;        }
h2 {font: bold 16px verdana;      color: #003BA1; }
h3 {font: bold 14px verdana;      color: #003BA1; }
h4 {font: bold 14px verdana;      color: #003BA1; font-style: italic; }
h5 {font: bold 12px verdana;      color: #003BA1; font-style: italic; }
h6 {font: 11px verdana;           color: #003BA1; font-style: italic;}
```

```
/**
* configuration for the boxes in the middle,
* clear: declare whether floating elements are forbidden to either side
* configure the width and location, and in addition the maximum size for
* the element on the right size and what happens if the element is overflowed
*/
#left, #right { clear:none }
#left { width:15em;float:left }
#right { height:auto; width:75%;float:left; overflow:scroll; max-height: 790px }

/**
*declaration of the background and what kind of border is around the elements
*/
#head, #left, #right, #foot {
   background:#eee;
   border:1px solid #aaa }

/**
* configuration of the list and what kind of font types are
*/
#left ul, li { margin:4px;list-style-type:none }
body { font: normal 12px verdana, tahoma, sans-serif }

/* ------------------------------------------------ */
/***************** menu coding *****************/
dl, dt, dd, ul, li {
            margin: 0;
            padding: 0;
            list-style-type: none;
}

/**
* location of the menu
*/
#menu {    width:15em; float:left; }

/**
* menu outlook
*/
#menu dt {
            cursor: pointer;
            margin: 2px 0;
            height: 20px;
            line-height: 20px;
            text-align: center;
            font-weight: bold;
            border: 1px solid gray;
            background: #ccc;
}

#menu dd { border: 1px solid gray; }
#menu li { text-align: center; background: #fff; }

#menu li a, #menu dt a {
            color: #000;
            text-decoration: none;
            display: block;
            border: 0 none;
            height: 100%;
}

#menu li a:hover, #menu dt a:hover { background: #eee; }
```

```
.mentions {
		position: absolute;
		top : 300px;
		left : 10px;
		color: #000;
		background-color: #ddd;
}

</style>

<!--
		when the menu calls this is what happens. the user clicks on something (say 'Registration'
		the menu opens and shows what is under 'Registration' and is open until the user clicks something
		else
-->
<script type="text/javascript">
		window.onload=sedill;
		function sedill(id) {
			var d = document.getElementById(id);
				for (var i = 1; i<=10; i++) {
					if (document.getElementById('smenu'+i))
{document.getElementById('smenu'+i).style.display='none';}
				}
			if (d) {d.style.display='block';}
		}
</script>

</head>
<body>
			<div id="head">
			  <p><h6><?php echo date('D dS \of M Y H:i:s A'); ?></h6></p>
			</div>
			<div id="left">
				<dl id="menu">

<dt onclick="javascript:sedill();"><a href="index.php">Home</a></dt>

<dt onclick="javascript:sedill('smenu2');">Registration</dt>
	<dd id="smenu2">
		<ul>
			<li><a
href="registration.php">Registration</a></li>
			<li><a href="line.php">Linage</a></li>
			<li><a
href="newcolor.php">Colour</a></li>
			<li><a
href="personnel.php">Personnel</a></li>
		</ul>
	</dd>

<dt onclick="javascript:sedill('smenu3');">Reports</dt>
	<dd id="smenu3">
		<ul>
			<li><a
href="mainReport.php">Reports</a></li>
			<li><a
href="search.php">Search</a></li>
			<li><a href="check.php">Take a
look</a></li>
		</ul>
	</dd>
```

```
                                    <dt onclick="javascript:sedill();"><a href="photo.php">Photos</a></dt>
                                    <dt onclick="javascript:sedill();">a href="usermanual.pdf" target="_blank">User
Manual</a></dt>

                                    <dt onclick="javascript:sedill();"><a href="about.php">About</a></dt>

                                    </dl>
                        </div>
                        <div id="right">
    <!-- break for content -->
```

3. footer.php

```
<!--
        Author: Ólafur Hannesson
        Date: (final changes and commenting) 05.04.2008
        Project: part of final year project autumn semester 2007, spring semester 2008
        Name of file: footer.php
        Connected files: 3rd part of the front page
        Function: the file holds the footer of the page
-->

    <!-- end of content -->
    </div>
    <div id="foot">
      <p class="sright">Copyright © 2007-2008 <a href="mailto:olafurhannesson@gmail.com">Ólafur Hannesson</a>. All rights
reserved.<br /><br /></p>
    </div>
</body>
</html>
```

4. index.php

```
<?php include 'main.php';?>
<!--
        Author: Ólafur Hannesson
        Date: (final changes and commenting) 05.04.2008
        Project: part of final year project autumn semester 2007, spring semester 2008
        Name of file: index.php
        Connected files: 2nd part of the front page, main.php (the outlook of the page and menu), footer.php (the footer of
the page)
        Function: this is the front page of the project
-->

        <p><h1><center>Welcome on the hompage of</center></h1></p>
                                        <p><center><img src="main1.jpg" border="0" width="500" height="602"
align="absmiddle"></center><br /></p>

<?php include 'footer.php' ?>
```

5. registration.php

```
<?php
        include 'main.php';
        include 'db.inc.php';
?>
<!--
        Author: Ólafur Hannesson
        Date: (final changes and commenting) 05.04.2008
        Project: part of final year project autumn semester 2007, spring semester 2008
        Name of file: registration.php
        Connected files: hestur.php (the view form), main.php (the outlook of the page and menu)
                footer.php (the footer of the page) db.inc.php (the connection to the database)
        Function: to allow the users to enter the data for a horse and upon submitting to view the registration
                before sending it to the database
-->

<!--
        this file displays some emtpy textboxes and a drop down lists with data gotten from the database.
        the user fills out the textboxes at will, and chooses data from the drop down menu if he so chooses.
        upon submitting the user will be able to view the registration before sending it to the database.
-->

<p><h1>Skráning</h1></p>
<ul>
<form action="hestur.php" method="post">
<table cellspacing="10px" align="left">
        <tr>
   <td>Registration date:</td>
   <td><input type="text" name="day" value="<?php echo date("m.d.y"); ?>"></td>
   <td>Price:</td>
   <td><input type="text" size="26" maxlength="256" name="price"></td>
  </tr>
        <tr>
   <td>ID number:</td>
   <td><input type="text" size="26" maxlength="256" name="fnumber"></td>
  </tr>
  <tr>
   <td>Name:</td>
   <td><input type="text" size="26" maxlength="256" name="name"></td>
  </tr>
  <tr>
   <td>Colour:</td>
   <td>
                <?php
                //get the colours from the database
                        $colour = @mysql_query('SELECT id, litur FROM litir');
                        if (!$colour) {
                           exit('<p>Unable to obtain colour list from the database.</p>');
                }
                ?>

        <select name="lid[]" size="1">
        <option selected value=" ">Select one colour</option>
        <option value=" ">----------</option>
                <?php
                // put them into a drop down list
                        while ($lit = mysql_fetch_array($colour)) {
                           $lname = $lit['litur'];
                           echo "<option value='$lname'>$lname</option>\n";
                        }
                ?>
```

```
                    </select>
                    </td>
  </tr>
  <tr>
   <td>Fathers ID number:</td>
   <td><input type="text" size="26" maxlength="256" name="ffnumber"></td>
  </tr>
        <tr>
   <td>Name of father:</td>
   <td><input type="text" size="26" maxlength="256" name="fname"></td>
  </tr>
  <tr>
   <td>Mothers ID number:</td>
   <td><input type="text" size="26" maxlength="256" name="mfnumber"></td>
  </tr>
        <tr>
   <td>Name of mother:</td>
   <td><input type="text" size="26" maxlength="256" name="mname"></td>
  </tr>
  <tr>
   <td>Personnel:</td>
   <td>
                    <?php
                    //get the personnel from the database
                        $pers = @mysql_query('SELECT id, name FROM personnel');
                        if (!$pers) {
                           exit('<p>Unable to obtain colour list from the database.</p>');
                    }
                    ?>

          <select name="person[]" size="1">
          <option selected value=" ">Personnel</option>
          <option value=" ">----------</option>
                    <?php
                    // put them into a drop down list
                        while ($personal = mysql_fetch_array($pers)) {
                           $pname = $personal['name'];
                           echo "<option value='$pname'>$pname</option>\n";
                        }
                    ?>

                    </select>
                    </td>
  </tr>
  <tr>
   <td>Location:</td>
   <td><input type="text" size="26" maxlength="256" name="location"></td>
  </tr>
  <tr>
   <td>Notes:</td>
   <td><textarea name="notes" rows="5" cols="30"></textarea></td>
  </tr>
</table>
 <p><!-- either clear the form or submit it to view it before sending it into the database -->
<input type="submit" value="Submit" name="submit" />
<input type="reset" value="Clear" name="clear" /></p>

<table align="right">
        <th>Father</th>

        <tr>
```

```
<td>
                <?php
                //get the horse from the database
                        $fath = @mysql_query('SELECT fnumber, name FROM father');
                        if (!$fath) {
                           exit('<p>Unable to obtain horse list from the database.</p>');
                }
                ?>

         <select name="fat[]" size="10">
                <?php
                // put them into a drop down list
                        while ($horselistF = mysql_fetch_array($fath)) {
                           $lfnumb = $horselistF['fnumber'];
                           $lname = $horselistF['name'];
                           echo "<option value='$lname'>$lfnumb - $lname</option>\n";
                        }
                ?>

                </select>
                </td>
        </tr>
        <th>Mother</th>
        <tr>
                <td>
                <?php
                //get the horse from the database
                        $moth = @mysql_query('SELECT fnumber, name FROM mother');
                        if (!$moth) {
                           exit('<p>Unable to obtain colour list from the database.</p>');
                }
                ?>

         <select name="mot[]" size="10">
                <?php
                // put them into a drop down list
                        while ($horselistM = mysql_fetch_array($moth)) {
                           $lmnumb = $horselistM['fnumber'];
                           $lmname = $horselistM['name'];
                           echo "<option value='$lmname'>$lmnumb - $lmname</option>\n";
                        }
                ?>

                </select>
                </td>
        </tr>
</table>




</form>
</ul>


<p><br /></p>


<?php include 'footer.php' ?>
```

6. hestur.php

```
<?php
		include 'main.php';
		include 'db.inc.php';
?>
<!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: hestur.php
		Connected files: registration.php (the registration form), main.php (the outlook of the page and menu), footer.php
(the footer of the page)
                                                db.inc.php (the connection to the database), check.php (the complete list
of horses in database)
		Function: to print out the data entered by the user, for the user to double check before sending it to the database
-->

<!--
		the file gets parameters from the form in registration.php, prints them out and allows the user to view them. Upon
		submitting, the user will be directed to check.php and can there see all the horses that are in the database
-->

<p><h1>Skráning</h1></p>
<ul>
<form action="check.php" method="post" enctype="multipart/form-data">
<table cellspacing="10px">
		<tr>
	<td>Registration date:</td>
	<td><?php echo $_POST['day']; ?></td>
	<td>Price:</td>
	<td><?php echo $_POST['price']; ?></td>
     </tr>
		<tr>
   <td>ID number:</td>
   <td><?php echo $_POST['fnumber']; ?></td>
  </tr>
  <tr>
   <td>Name:</td>
   <td><?php echo $_POST['name']; ?></td>
  </tr>
  <tr>
   <td>Colour:</td>
   <td>
		<?php
		$color = '';
		//gets the data chosen in the drop down list and determines if the variable is an array
			if(is_array($_POST['lid'])) {
				//looping through the values in the array
					foreach($_POST['lid'] as $color) {
						//printing out the name of the colour
						echo "$color<br />";
					}
				} else {
					echo 'No colour was selected';

				}
			?>
			</td>
  </tr>
  <tr>
   <td>Fathers ID number:</td>
```

```
  <td><?php echo $_POST['ffnumber']; ?></td>
 </tr>
        <tr>
  <td>Name of father:</td>
  <td><?php echo $_POST['fname']; ?></td>
 </tr>
 <tr>
  <td>Mothers ID number:</td>
  <td><?php echo $_POST['mfnumber']; ?></td>
 </tr>
        <tr>
  <td>Name of mother:</td>
  <td><?php echo $_POST['mname']; ?></td>
 </tr>
 <tr>
  <td>Personnel:</td>
  <td>
         <?php
         $person = '';
         //gets the data chosen in the drop down list and determines if the variable is an array
                if(is_array($_POST['person'])) {
                        //looping through the values in the array
                                foreach($_POST['person'] as $person) {
                                        echo "$person<br />";
                                }
                        } else {
                                echo 'No person was selected';

                        }
                ?>
                </td>
 </tr>
 <tr>
  <td>Location:</td>
  <td><?php echo $_POST['location']; ?></td>
 </tr>
 <tr>
  <td>Notes:</td>
  <td><?php echo $_POST['notes']; ?></td>
 </tr>

        </table>

        <p><input type="submit" value="Submit" name="submitHorseData" /></p>
        </form>
</ul>

<p><br /></p>
<?php
//first check if the variable is set, if it is not set, the user must go back and enter
//the ´fnumber´ because it will return error messages
        if(isset($_POST['fnumber'])) {
                $day = $_POST['day'];
                $price = $_POST['price'];
                $lid = $color;
                $fnumber = $_POST['fnumber'];
                $name = $_POST['name'];
                $ffnumber = $_POST['ffnumber'];
                $fname = $_POST['fname'];
                $mfnumber = $_POST['mfnumber'];
                $mnumber = $_POST['mnumber'];
                $mname = $_POST['mname'];
```

```
                $care = $person;
                $location = $_POST['location'];
                $notes = $_POST['notes'];

                if ($fnumber == '') {
                        exit('<p>You must fill in the ID number</p>');
                }

                //puts the data into variable
                $sql1= "INSERT INTO hestur SET
                                day = now(),
                                fnumber = '$fnumber',
                                name = '$name',
                                ffnumber = '$ffnumber',
                                fname = '$fname',
                                mfnumber = '$mfnumber',
                                mname = '$mname',
                                care = '$care',
                                litirid = '$lid',
                                location = '$location',
                                notes = '$notes',
                                price = '$price'";


                //determines if the query is ok    and sends it
                if(mysql_query($sql1)) {
                        echo '<a href="check.php">';
                } else {
                        //..or gives out an error message
                        echo '<p>: ' . mysql_error() . '</p>';
                }
        }
?>

<p><br /></p>

<?php include 'footer.php' ?>
```

7. newcolor.php

```
<?php
		include 'main.php';
		include 'db.inc.php';
?>
<!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: newcolor.php
		Connected files: registration.php (the registration form), main.php (the outlook of the page and menu), footer.php
(the footer of the page)
                                                db.inc.php (the connection to the database)
		Function: to keep a list of all colours the user creates
-->

<!--
		the file allows the user to register any colours and keep a data about them.
-->

<!-- the form the user uses to register any new colours -->
<p><h1>Register new colour</h1></p>
<p> </p>
<ul>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
		<table cellspacing="10px">
	<tr>
	  <td><h3>Name of colour:</h3>
	  <td><input type="text" size="26" maxlength="256" name="litur">
	</tr>
			<tr>
	  <td><h3>Description:</h3>
	  <td><textarea name="description" rows="5" cols="42"></textarea>
	</tr>
		</table>
		<input type="submit" value="Submit" name="submit" />
		<input type="reset" value="Clear" name="clear" />
</form>

<?php
// if the name of the colour has been set
if(isset($_POST['litur'])) {
		// get the input
		$litur = $_POST['litur'];
		$description = $_POST['description'];
		//create the query
		$sql= "INSERT INTO litir SET litur = '$litur', description = '$description'";
		// send the query
		if(mysql_query($sql)) {
			//print if successful
			echo '<p>Your colour has been added to the database!</p>';
			echo '<p>Please click <a href="newcolor.php">here</a> to return!</p>';
		} else {
			//print if failed
			echo '<p>There is a problem: ' . mysql_error() . '</p>';
		}
}
?>


</ul>
```

```
<br />

<?php
//define the variable 'action' and get it
if(isset($_GET['action'])) {
	$action = $_GET['action'];
} else {
	$action = 'bugl';
}


if($action == '') {
	echo " ";
} elseif($action == 'del' and isset($_GET['id'])) {
	// if the action is 'del' and the id is set
	$id = $_GET['id'];

	// create a query
	$sqlCol = "DELETE FROM litir WHERE id = '$id'";
	//send the query
	if(mysql_query($sqlCol)) {
		//if the query is successful
		echo '<p>Your colour has been deleted from the database!</p>';
		echo '<p>Please click <a href="newcolor.php">here</a> to return!</p>';
	} else {
		//if the query is not successful
		echo '<p>There is a problem: ' . mysql_error() . '</p>';
	}


//	header('location: ' . $_SERVER['PHP_SELF']);
	exit();
}
?>


<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<p>The following colours are registered in the database</p>
<table>
<tr>
	<th>ID number</th>
	<th>Name </th>
	<th></th>
</tr>


<?php
//get all data from the table ´litir'
$sqlC = 'SELECT * FROM litir';
//send the query
$list = mysql_query($sqlC);
//check the results of the query
if(!$list) {
	//if they are not good, print an error message
	exit('Database error: ' . mysql_error());
}

//check if the number of rows is empty
if(mysql_num_rows($list) > 0) {
	// while there is data in the resulting array
	while($l = mysql_fetch_array($list)) {
		//...print out this table
```

```
                        ?>
                <tr valign="top">
                        <td><?php echo $l['id']; ?></td>
                        <td><?php echo $l['litur']; ?></td>
                        <!-- if the user wants to delete from the database -->
                        <td>[<a href="<?php echo $_SERVER['PHP_SELF']; ?>?action=del&id=<?php echo $l['id']; ?>"
onclick="return confirm('Delete this colour from the database?');">Delete</a> ]
 </td>
</tr>
<?php
                }
} else {
                ?>
                <tr><td colspan="3">No colours in database.</td></tr>
                <?php
}
?>
</form>
</table>



<?php include 'footer.php' ?>
```

8. personnel.php

```
<?php
		include 'main.php';
		include 'db.inc.php';
?>
<!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: personnel.php
		Connected files: registration.php (the registration form), main.php (the outlook of the page and menu), footer.php
(the footer of the page)  db.inc.php (the connection to the database), check.php (the complete list of horses in database)
		Function: to register any personnel that is are needed around the horses
-->
<!--
		the file allows the user to register any person and keep a data about them.
-->
<!-- the form the user must fill in -->
<p><h1>The following personnel have been inserted into the database: </h1></p>
<ul>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<table cellspacing="10px">
  <tr>
   <td>Name:
   <td><input type="text" size="26" maxlength="256" name="name">
  </tr>
  <tr>
   <td>Address:
   <td><input type="text" size="26" maxlength="256" name="address">
  </tr>
  <tr>
   <td>Occupation:
   <td><input type="text" size="26" maxlength="256" name="occupation">
  </tr>
  <tr>
   <td>Email address:
   <td><input type="text" size="26" maxlength="256" name="email">
  </tr>
		<tr>
   <td>Phone:
   <td><input type="text" size="26" maxlength="256" name="phone">
  </tr>
  <tr>
   <td>GSM:
   <td><input type="text" size="26" maxlength="256" name="gsm">
  </tr>
  <tr>
   <td>Notes:
   <td><input type="text" size="26" maxlength="256" name="notes">
  </tr>
</table>
<p>
		<input type="submit" value="Submit" name="submit" />
		<input type="reset" value="Clear" name="clear" /></p>
		</form>
</ul>
<p><br /></p>
<?php
// to see if the user has filled in the name which is essential for this form
		if(isset($_POST['name'])) {
			//get the variables
```

```
                        $name = $_POST['name'];
                        $address = $_POST['address'];
                        $occupation = $_POST['occupation'];
                        $email = $_POST['email'];
                        $phone = $_POST['phone'];
                        $gsm = $_POST['gsm'];
                        $notes = $_POST['notes'];

                        // if the name is empty, print out a message
                        if ($name == '') {
                                exit('<p>You must put a name!! Please press the <BACK> button on your browser to fix
it.</p>');
                        }

                        // create a query
                        $sqlP = "INSERT INTO personnel SET
                                                name = '$name',
                                                address = '$address',
                                                occupation = '$occupation',
                                                email = '$email',
                                                phone = '$phone',
                                                gsm = '$gsm',
                                                notes = '$notes'";


                        // send the query
                        if(mysql_query($sqlP)) {
                                // print a message if the query was successful
                                echo 'You have registered a personnel! Please click <a href="personnel.php">here</a> to
return or in the menu to the side.';
                        } else {
                                //..or print an error message
                                echo '<p>There are problems: ' . mysql_error() . '</p>';
                        }
                }
?>

<!-- table that holds the information about the personnel existing in the db -->
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<table>
<tr>
                <th>  </th>
                <th>  </th>
                <th>Name</th>
                <th>Address </th>
                <th>Occupation </th>
                <th>email </th>
                <th>GSM </th>
                <th>phone</th>
                <th>notes</th>
</tr>

<?php
//define the variable 'action' and get it
if(isset($_GET['action'])) {
                $action = $_GET['action'];
} else {
                $action = 'bugl';
}
// if the variable holds 'edit'
if($action == 'edit') {
                echo "edit";
```

```
} elseif($action == 'del' and isset($_GET['id'])) {
		//if the variable holds 'del' and the id is set
		$id = $_GET['id'];

		// create query
		$sqlPerson = "DELETE FROM personnel WHERE id = '$id'";
		// send the query
		if(mysql_query($sqlPerson)) {
			// if the query was successful,
			echo '<p>The person has been successfully deleted from the database!</p>';
			echo '<p>Press <a href="personnel.php">here</a> to return!</p>';
		} else {
			// if there was an error
			echo '<p>There is a problem: ' . mysql_error() . '</p>';
		}

//		header('location: ' . $_SERVER['PHP_SELF']);
// all messages have been printed, the script is terminated
		exit();
}
// get all teh data from personnel
$sql1 = 'SELECT * FROM personnel';
// send the query
$list = mysql_query($sql1);
// check if it is alright with the results
if(!$list) {
		//... else print an error
		exit('Database error: ' . mysql_error());
}

//check if the results return empty rows
if(mysql_num_rows($list) > 0) {
		// while there is any data in the resulting array..
		while($l = mysql_fetch_array($list)) {
			//..print it out in a table
			?>
		<tr valign="top">
			<td></td>
			<td></td>
			<td><?php echo $l['name']; ?>.</td>
			<td><?php echo $l['address']; ?></td>
			<td><?php echo $l['occupation']; ?></td>
			<td><?php echo $l['email']; ?></td>
			<td><?php echo $l['phone']; ?></td>
			<td><?php echo $l['gsm']; ?></td>
			<td><?php echo $l['notes']; ?></td>
			<!-- if the user wants to delete a person from the database, seeks confirmation -->
			<td>[ Edit |
		<a href="<?php echo $_SERVER['PHP_SELF']; ?>?action=del&id=<?php echo $l['id']; ?>" onclick="return
confirm('Delete this personnel from the database?');">Delete</a> ]</td>
		<td><br /></td>
</tr>
<?php
		}
} else {
		?>
		<tr><td colspan="3">No persons in the database.</td></tr>
		<?php
}
?>

<?php include 'footer.php' ?>
```

9. line.php

```
<?php
        include 'main.php';
        include 'db.inc.php';
?>
<!--
        Author: Ólafur Hannesson
        Date: (final changes and commenting) 05.04.2008
        Project: part of final year project autumn semester 2007, spring semester 2008
        Name of file: line.php
        Connected files: registration.php (the registration form), main.php (the outlook of the page and menu),
                                                footer.php (the footer of the page)
                                                db.inc.php (the connection to the database)
                                                parent.php (confirmation page)
        Function: to keep list of the information about the parents of horse
-->

<!--
        the file allows the user to register any horse that is frequently used as parent for any horse
-->
<p><h1>Linage: </h1></p>
<ul>
<form action="parent.php" method="post">
<table cellspacing="10px">
        <tr><td>
        <fieldset><legend>Gender</legend>
        <select size="1" name="gender[]">
                <option selected value=" ">Select gender</option>
                <option value=" ">----------</option>
                <option value="Male">Male</option>
                <option value="Female">Female</option>
        </select>
        </fieldset>
        </td>
        <td>    </td>
        </tr>
  <tr>
   <td>ID number:
   <td><input type="text" size="26" maxlength="255" name="fnumber">
  </tr>
        <tr>
   <td>Name:
   <td><input type="text" size="26" maxlength="255" name="name">
  </tr>
  <tr>
                <td>Colour:
                <td>
                <?php
                        $colour = @mysql_query('SELECT id, litur FROM litir');
                        if (!$colour) {
                           exit('<p>Unable to obtain colour list from the database.</p>');
                }
                ?>

                <select name="lid[]" size="1">
                <option selected value=" ">Select one colour</option>
                <option value=" ">----------</option>
                <?php
                        while ($lit = mysql_fetch_array($colour)) {
                           $lname = $lit['litur'];
                           echo "<option value='$lname'>$lname</option>\n";
```

```
                    } extract
                ?>

                </select>
                </tr>
                <tr>
                <td>Fathers ID number:
                <td><input type="text" size="26" maxlength="255" name="ffnumber">
                </tr>
                <tr>
                <td>Name of father:
                <td><input type="text" size="26" maxlength="255" name="fname">
                </tr>
                <tr>
                <td>Mothers ID number:
                <td><input type="text" size="26" maxlength="255" name="mfnumber">
                </tr>
                <tr>
                <td>Name of mother:
                <td><input type="text" size="26" maxlength="255" name="mname">
                </tr>
                <tr>
                <td>Owner:
                <td>
                <?php
                    $pers = @mysql_query('SELECT id, name FROM personnel');
                    if (!$pers) {
                       exit('<p>Unable to obtain colour list from the database.</p>');
                }
                ?>

                <select name="person[]" size="1">
                <option selected value=" ">Owner</option>
                <option value=" ">----------</option>
                <?php
                    while ($personal = mysql_fetch_array($pers)) {
                       $pname = $personal['name'];
                       echo "<option value='$pname'>$pname</option>\n";
                    }
                ?>

                </select>
                </tr>
                <tr>
                <td>Location:
                <td><input type="text" size="26" maxlength="255" name="location">
                </tr>
                <tr>
                <td>Contact:
                <td><input type="text" size="26" maxlength="255" name="phone">
                </tr>
                </table>
                <p>
                <input type="submit" value="Submit" name="submit" />
                <input type="reset" value="Clear" name="clear" /></p>
</form>
</ul>
<p><br /></p>

<!-- Male horses: table holding informations -->
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<table border="1" cellpadding="5" cellspacing="0" bordercolor="#000000">
```

```
<tr>
		<th>Gender</th>
		<th>ID number</th>
		<th>Name</th>
		<th>Owner </th>
		<th>Location</th>
</tr>

<?php
// check if the variable 'action' is set and get it
if(isset($_GET['action'])) {
		$action = $_GET['action'];
} else {
		$action = '';
}
// if the data in the variable is 'edit'
if($action == 'edit') {
		echo "";
} elseif($action == 'del' and isset($_GET['id'])) {
		// if the data in the variable is 'del' and 'id' is set
		// get the id
		$id = $_GET['id'];

		// create the query
		$sqlDel = "DELETE FROM father WHERE id = '$id'";
		// send the query and print results
		if(mysql_query($sqlDel)) {
			echo '<p>Your horse has been successfully deleted from the database!</p>';
			echo '<p>Press <a href="line.php">here</a> to return!</p>';
		} else {
			// or print error message
			echo '<p>There is a problem: ' . mysql_error() . '</p>';
		}


//		header('location: ' . $_SERVER['PHP_SELF']);
		// the messages have been output and the script is terminated
		exit();
}

// get all the data from table 'father'
$sqlFather = 'SELECT * FROM father';
//send the query
$listMe = mysql_query($sqlFather);
//if the query doesn´t return results, print out error message
if(!$listMe) {
		exit('Database error: ' . mysql_error());
}

// if the number of rows is null
if(mysql_num_rows($listMe) > 0) {
		// get the result row as array and return to variable
		while($m = mysql_fetch_array($listMe)) {
			// get the value from the array and print the results in a table
			?>
		<tr valign="top">
			<td><?php echo $m['gender']; ?></td>
			<td><?php echo $m['fnumber']; ?></td>
			<td><?php echo $m['name']; ?></td>
			<td><?php echo $m['ownerid']; ?></td>
			<td><?php echo $m['location']; ?></td>
			<td>[ Edit |
```

```
		<a href="<?php echo $_SERVER['PHP_SELF']; ?>?action=del&id=<?php echo $m['id']; ?>" onclick="return
confirm('Delete this personnel from the database?');">Delete</a> ]</td>
		<td><br /></td>
</tr>

<?php
		}
} else {
		//if the number of rows is null, print out message
		?>
		<tr><td colspan="3">No horses</td></tr>
		<?php
}
?>
</table>

<br /><br />

<!-- Female horses: table holding informations -->
<table border="1" cellpadding="5" cellspacing="0" bordercolor="#000000">
<tr>
		<th>Gender</th>
		<th>ID number</th>
		<th>Name</th>
		<th>Owner </th>
		<th>Location</th>
</tr>

<?php
// check if the variable 'action' is set and get it
if(isset($_GET['action'])) {
		$action = $_GET['action'];
} else {
		$action = '';
}

// if the data in the variable is 'edit'
if($action == 'edit') {
		echo "";
} elseif($action == 'del' and isset($_GET['id'])) {
		// if the variable is 'del' and the id is set
		//get the id
		$id = $_GET['id'];

		//create a query
		$sqlDelM = "DELETE FROM mother WHERE id = '$id'";
		//send the query and print message if successful
		if(mysql_query($sqlDelM)) {
				echo '<p>Your horse has been successfully deleted from the database!</p>';
				echo '<p>Press <a href="line.php">here</a> to return!</p>';
		} else {
				//print out error message
				echo '<p>There is a problem: ' . mysql_error() . '</p>';
		}

//		header('location: ' . $_SERVER['PHP_SELF']);
		// the messages have been output and the script is terminated
		exit();
}

//get all the data from the table 'mother'
$sqlf = 'SELECT * FROM mother';
```

```
//send the query
$listFe = @mysql_query($sqlf);
//if the query doesn´t return results, print out error message
if(!$listFe) {
	exit('Database error: ' . mysql_error());
}

// if the number of rows is null
if(mysql_num_rows($listFe) > 0) {
	// get the result row as array and return to variable
	while($f = mysql_fetch_array($listFe)) {
		// get the value from the array and print the results in a table
		?>
	<tr valign="top">
		<td><?php echo $f['gender']; ?></td>
		<td><?php echo $f['fnumber']; ?></td>
		<td><?php echo $f['name']; ?></td>
		<td><?php echo $f['ownerid']; ?></td>
		<td><?php echo $f['location']; ?></td>
		<td>[ Edit |
	<a href="<?php echo $_SERVER['PHP_SELF']; ?>?action=del&id=<?php echo $f['id']; ?>" onclick="return
confirm('Delete this personnel from the database?');">Delete</a> ]</td>
	<td><br /></td>
</tr>

<?php
	}
} else {
	//print out message that it is empty
	?>
	<tr><td colspan="3">No horses</td></tr>
	<?php
}
?>
</table>
</form>
<?php include 'footer.php' ?>
```

10. parent.php

```
<?php
include 'main.php';
include 'db.inc.php';
?>
<!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: parent.php
		Connected files: main.php (the outlook of the page and menu),
								footer.php (the footer of the page)
								db.inc.php (the connection to the database)
								line.php (registration page)
		Function: to keep list of the information about the parents of horse
-->

<!--
		the file allows the user to view his registration before submitting it to the database
-->

<p><h1>Skráning</h1></p>
<!-- this form show all the information -->
<form action="line.php" method="post">
<table cellspacing="10px">
		<tr>
    <td>Gender:</td>
    <td>
    <?php
    //checks if the the data sent is an array, iterates through it and puts it into variable
			if(is_array($_POST['gender'])) {
				foreach($_POST['gender'] as $g) {
					echo "$g<br />";
				}
			} else {
				// no gender was selected
				echo 'No gender was selected';

			}
		?>
		</td>
  </tr>
		<tr>
   <td>ID number:</td>
   <td><?php echo $_POST['fnumber']; ?></td>
  </tr>
  <tr>
   <td>Name:</td>
   <td><?php echo $_POST['name']; ?></td>
  </tr>
  <tr>
   <td>Colour:</td>
   <td>
   <?php
   // checks if the data is a variable, iterates through it and sends to an variable
			if(is_array($_POST['lid'])) {
				foreach($_POST['lid'] as $c) {
					echo "$c<br />";
				}
			} else {
				echo 'No colour was selected';
```

```
                                        }
                        ?>
                        </td>
  </tr>
  <tr>
   <td>Fathers ID number:</td>
   <td><?php echo $_POST['ffnumber']; ?></td>
  </tr>
          <tr>
   <td>Name of father:</td>
   <td><?php echo $_POST['fname']; ?></td>
  </tr>
  <tr>
   <td>Mothers ID number:</td>
   <td><?php echo $_POST['mfnumber']; ?></td>
  </tr>
          <tr>
   <td>Name of mother:</td>
   <td><?php echo $_POST['mname']; ?></td>
  </tr>
  <tr>
   <td>Personnel:</td>
   <td>
            <?php
            // checks if data is an array, iterates through it and sends to a variable
                        if(is_array($_POST['person'])) {
                                        foreach($_POST['person'] as $p) {
                                                    echo "$p<br />";
                                        }
                        } else {
                                        echo 'No person was selected';

                                        }
                        ?>
                        </td>
  </tr>
  <tr>
   <td>Location:</td>
   <td><?php echo $_POST['location']; ?></td>
  </tr>
  <tr>
   <td>Notes:</td>
   <td><?php echo $_POST['phone']; ?></td>
  </tr>
</table>
<input type="submit" value="Submit" name="submit" />
</form>

<?php
//checks if the data for fnumber has been set
if(isset($_POST['fnumber'])) {
            //gets the data and puts it into variables
            $lid = $c;
            $gender = $g;
            $fnumber = $_POST['fnumber'];
            $name = $_POST['name'];
            $ffnumber = $_POST['ffnumber'];
            $fname = $_POST['fname'];
            $mfnumber = $_POST['mfnumber'];
            $mname = $_POST['mname'];
            $care = $p;
```

```
$location = $_POST['location'];
$phone = $_POST['phone'];

// if fnumber is empty,
if ($fnumber == '') {
	// an error messages are printed out
	exit('<p>You must fill in the ID number</p>');
}

// if the gender is male...
if ($gender == 'Male') {
	// a query is created to send this into a correct table
	$sql1= "INSERT INTO father SET
				fnumber = '$fnumber',
				name = '$name',
				gender = '$gender',
				colour = '$lid',
				ffnumber = '$ffnumber',
				fname = '$fname',
				mfnumber = '$mfnumber',
				mname = '$mname',
				ownerid = '$care',
				phone = '$phone',
				location = '$location'";


	// sends the query and checks if it is good
	if(mysql_query($sql1)) {
		//then sends you to the correct page
		echo '<a href="line.php';
	} else {
		//or prints out an error message
		echo '<p>There are problems: ' . mysql_error() . '</p>';
	}
} elseif($gender == 'Female'){
	//if the gender is female
	// create a query to send it into a correct table
	$sql1= "INSERT INTO mother SET
				fnumber = '$fnumber',
				name = '$name',
				gender = '$gender',
				colour = '$lid',
				ffnumber = '$ffnumber',
				fname = '$fname',
				mfnumber = '$mfnumber',
				mname = '$mname',
				ownerid = '$care',
				phone = '$phone',
				location = '$location'";

	//sends the query and checks if it is good
	if(mysql_query($sql1)) {
		// then sends you to another page
		echo '<a href="line.php';
	} else {
		//or prints error message
		echo '<p>There are problems: ' . mysql_error() . '</p>';
	}
} else {
	// if no gender is chosen, an error message are printed
	echo '<p>There is a problem with the data insertion. Please make sure you have chosen a gender and try
again.</p>';
```

```
            }
}
?>
<form method="get" enctype="text/plain">
<input type="hidden" value="hvað" name="gete">
</form>


<?php include 'footer.php' ?>
```

11. mainReport.php

```
<?php
	include 'main.php';
	include 'db.inc.php';
?>

<!--
	Author: Ólafur Hannesson
	Date: (final changes and commenting) 05.05.2008
	Project: part of final year project autumn semester 2007, spring semester 2008
	Name of file: mainReport.php
	Connected files: main.php (the outlook of the page and menu), footer.php (the footer of the page)
					db.inc.php (the connection to the database), reports.php (the report)
					excel.php (table format for excel), wordi.php (table format for word documents)
	Function: to show a list of all horses, and allow the user to pick a horse or to get a report in
				Word format or Excel format to view all the horses
-->

<!--
	the file allows the user to fill in 3 text areas, input being the name of the show if given, location and date.
	the user then picks the horses he wants to see by marking in the checkboxes and the group.
-->


<!-- the form in which the user picks a restriction to order by for html form -->
<form method="post" action="reportsN.php" target="_blank">
	<table border="0" cellpadding="0" cellspacing="0">
	<tr>
		<td><h4>Get report in html: </h4></td>
		<td>
			<p align="center">
			<!-- the fields to order by in -->
			<select name="fields[]" size="1">
			<option value=" ">Pick a restriction</option>
			<option value="fnumber">ID Number</option>
			<option value="name">Name</option>
			<option value="ffnumber">Fathers ID Number</option>
			<option value="fname">Fathers name</option>
			<option value="mfnumber">Mothers ID Number</option>
			<option value="mname">Mothers name</option>
			<option value="location">Location</option>
			<option value="day">Registration day</option>
			<option value="price">Price</option>
			</select>
		</td>
		<td>Get horse list: <input type="submit" value="Listing....." name="Listing"></p></td>
	</tr>
</form>

<!-- the form in which the user picks a restriction to order by for html form -->
<form method="post" action="wordi.php" target="_blank">
	<tr>
		<td><h4>Get report in a Word document: </h4></td>
		<td>
			<p align="center">
			<!-- the fields to order by in -->
			<select name="fields[]" size="1">
			<option value=" ">Pick a restriction</option>
			<option value="fnumber">ID Number</option>
			<option value="name">Name</option>
			<option value="ffnumber">Fathers ID Number</option>
			<option value="fname">Fathers name</option>
			<option value="mfnumber">Mothers ID Number</option>
			<option value="mname">Mothers name</option>
			<option value="location">Location</option>
			<option value="day">Registration day</option>
			<option value="price">Price</option>
			</select>
		</td>
```

```
                <td>Get horse list: <input type="submit" value="Listing....." name="Listing"></p></td>
        </tr>
</form>

<!-- the form in which the user picks a restriction to order by for html form -->
<form method="post" action="excel.php" target="_blank">
        <tr>
                <td><h4>Get report in a Excel spreadsheet: </h4></td>
                <td>
                        <p align="center">
                        <!-- the fields to order by in -->
                        <select name="fields[]" size="1">
                        <option value=" ">Pick a restriction</option>
                        <option value="fnumber">ID Number</option>
                        <option value="name">Name</option>
                        <option value="ffnumber">Fathers ID Number</option>
                        <option value="fname">Fathers name</option>
                        <option value="mfnumber">Mothers ID Number</option>
                        <option value="mname">Mothers name</option>
                        <option value="location">Location</option>
                        <option value="day">Registration day</option>
                        <option value="price">Price</option>
                        </select>
                </td>
                <td>Get horse list: <input type="submit" value="Listing....." name="Listing"></p></td>
        </tr>
        </table>
</form>

<form action="reports.php" method="post">
<p><h2>The following horses are registered in the database: </h2></p>
<table>
<tr>
<td>Please insert name of the show: </td>
<td><input type="text" size="26" maxlength="256" name="nameOfShow"></td>
<td>Please insert the location of the show: </td>
<td><input type="text" size="26" maxlength="256" name="locationOfShow"></td>
<td>Please insert the date of the show: </td>
<td><input type="text" size="26" maxlength="256" name="dateOfShow"></td>
</tr>
</table>
<table border="1" cellpadding="5" cellspacing="0">
<tr>
        <th>  </th>
        <th>  </th>
        <th> ID number</th>
        <th> Name </th>
        <th> Location </th>
        <th> Owner </th>
        <th></th>
        <th>Group</th>
</tr>
<?php
// get all the data from table 'hestur'
$sqlCheck = "SELECT * FROM hestur";
//send the query and get the results back
$list = mysql_query($sqlCheck);
//if the results are not good
if(!$list) {
        //..print out error message
        exit('Database error: ' . mysql_error());
}

//if the number of rows is not empty..
if(mysql_num_rows($list) > 0) {
        //get the data from the array
        while($l = mysql_fetch_array($list)) {
                //print out the data
                ?>
        <tr valign="top">
                <td></td>
```

```
                        <td><?php echo $l['id']; ?>.</td>
                        <td><?php echo $l['fnumber']; ?></td>
                        <td><?php echo $l['name']; ?></td>
                        <td><?php echo $l['location']; ?></td>
                        <td><?php echo $l['care']; ?></td>
                        <!-- the checkbox the user needs to mark if he wants it in the report -->
                <td><input type="checkbox" name="box[]" value="<?php echo $l['fnumber']; ?>"></td>
                <!--the text area the user needs to mark if he wants it to be in a order -->
                <td><input type="text" size="26" maxlength="256" name="whichGroup[]">
                        </td>
                </tr>
<?php
                }
} else {
                // if there are no horses in the database
                ?>
                <tr><td colspan="3">No horses</td></tr>
                <?php
}
?>
<tr><td></td><td></td><td><input type="submit" value="Submit" name="submit" /></td>
<td><input type="reset" value="Clear" name="clear" /></td></tr>
</form>
</table>
<br />
<p> </p>


<?php include 'footer.php' ?>
```

12. reports.php

```
<html>
<head>
<title>ReView: registration and viewing webpage</title>
</head>
<body>
		<p><h1><center>Reports</center></h1></p>

<!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: reports.php
		Connected files: db.inc.php (the connection to the database), mainReports.php (the reports page)
		Function: to show a list of all horses
-->

<!--
		given the information the user gave in mainReports.php, the user will see a list of horses
		with the information about them.
-->
<?php
		include 'db.inc.php';

// if the user pushed the 'submit' button
if (isset($_POST['submit'])) {
		//get the data from mainReports.php
   $box = $_POST["box"];
   $text = $_POST['whichGroup'];

   //print out the name of the show the user gave
   echo '<h2>'.$_POST['nameOfShow'].'</h2><br>';
   // print out the location and the date the user gave
   echo 'Location: ' . $_POST['locationOfShow'] . ' / date: ' . $_POST['dateOfShow'] . '<br>';
   //count the horses chosen
		$how_many = count($box);
		//print out the number of the horses chosen
   echo 'Number of horses chosen: '.$how_many.'<br>';
   //if the number of horses chosen is zero, warn the user and offer him to go back
   if ($how_many == 0) {
      echo 'You have not chosen any horses! Press <a href="mainReport.php">here</a> to go back<br>';
   }
   //declare variables
		$varCheckbox = '';
		$varText = '';

		// do the following loop for each horse chosen by the checkbox
		for ($i=0; $i<$how_many; $i++) {
			//get the value from the checkbox
			$varCheckbox = $box[$i];
			// get the value from the text box
			$varText = $text[$i];

			// create the query
			$s = "SELECT * FROM hestur WHERE fnumber = '$varCheckbox'";
			// send the query
			$list = mysql_query($s);
			// if the query is not good
			if(!$list) {
				// print an erro message
				exit('Database error: ' . mysql_error());
			}

			// check if the results return anything
			if(mysql_num_rows($list) > 0) {
				// send the data from an array to a variable
				while($l = mysql_fetch_array($list)) {
```

```
// print out all the data
?>
<table>
	<tr>
		<tr>
			<td><b><u>Nr. <?php echo ($i+1); ?> / Group <?php echo
$varText; ?></u></b></td>
			<td>    </td>
			<td>    </td>
			<td>    </td>
			<td>    </td>
			<td>Registration date: <?php echo $l['day']; ?></td>
		</tr>
		<tr><td> </td></tr>
		<tr>
			<td>ID Number: </td>
			<td><?php echo $l['fnumber']; ?></td>
			<td>    </td>
			<td>    </td>
			<td>    </td>
			<td>Price: <u><?php echo number_format($l['price'], 2, ',', ' ');
?></u></td>
		</tr>
		<tr>
			<td>Name: </td>
			<td><?php echo $l['name']; ?></td>
		</tr>
		<tr>
			<td>Colour: </td>
			<td><?php echo $l['litirid']; ?></td>
		</tr>
		<tr>
			<td>-----------</td>
			<td>-------------------------</td>
			<td>-------------------------</td>
		</tr>
		<tr>
			<td>Father </td>
			<td><?php echo $l['ffnumber']; ?></td>
			<td>    </td>
			<td>Name: </td>
			<td><?php echo $l['fname']; ?></td>
		</tr>
		<tr>
			<td>Mother </td>
			<td><?php echo $l['mfnumber']; ?></td>
			<td>    </td>
			<td>Name: </td>
			<td><?php echo $l['mname']; ?></td>
		</tr>
		<tr>
			<td>Location </td>
			<td><?php echo $l['location']; ?></td>
		</tr>
		<tr>
			<td>Owner </td>
			<td><?php echo $l['care']; ?></td>
		</tr>
		<!-- marks the line between horses -->

<tr>/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
/\/\/\/\/\/\/\</tr>
		<tr> </tr>
	</tr>
</table>

<?php

	}
}
```

```
            }
}
echo "<br><br>";

?>


</body>
</html>
```

13. reportsN.php

```
<html>
<head>
<title>ReView: registration and viewing webpage</title>
</head>
<body>
	<p><h1><center>Reports</center></h1></p>

<!--
	Author: Ólafur Hannesson
	Date: (final changes and commenting) 05.05.2008
	Project: part of final year project autumn semester 2007, spring semester 2008
	Name of file: reportsN.php
	Connected files: mainReport.php (restriction sent), db.inc.php (the connection to the database)
	Function: to show a list of all horses depending on restriction chosen by the user
	Output: a html form
-->
<?php
	include 'db.inc.php';

$fieldChosen = '';
//gets the data chosen in the drop down list and determines if the variable is an array
if(is_array($_POST['fields'])) {
	//looping through the values in the array
	foreach($_POST['fields'] as $fieldChosen) {
		//printing out the name of the colour
		$fields = $fieldChosen;
	}
} else {
	echo 'No restriction was selected';

}

//the query is sent to the database
$result = mysql_query("SELECT * FROM hestur ORDER BY $fields");
// if the query is not good
if(!$result) {
	// print an erro message
	exit('Database error: ' . mysql_error());
}

// check if the results return anything
if(mysql_num_rows($result) > 0) {
	// send the data from an array to a variable
	while($l = mysql_fetch_array($result)) {
		// print out all the data
		?>
		<table>
			<tr>
				<tr>
					<td><b><u>Nr. <?php echo $l['id']; ?></u></b></td>
					<td>    </td>
					<td>    </td>
					<td>    </td>
					<td>    </td>
					<td>Registration date: <?php echo $l['day']; ?></td>
				</tr>
				<tr><td> </td></tr>
				<tr>
					<td>ID Number: </td>
```

```
                        <td><?php echo $l['fnumber']; ?></td>
                        <td>    </td>
                        <td>    </td>
                        <td>    </td>
                        <td>Price: <u><?php echo number_format($l['price'], 2, ',', ' '); ?></u>
kr.</td>
                    </tr>
                    <tr>
                        <td>Name: </td>
                        <td><?php echo $l['name']; ?></td>
                    </tr>
                    <tr>
                        <td>Colour: </td>
                        <td><?php echo $l['litirid']; ?></td>
                    </tr>
                    <tr>
                        <td>-----------</td>
                        <td>-------------------------</td>
                        <td>-------------------------</td>
                    </tr>
                    <tr>
                        <td>Father </td>
                        <td><?php echo $l['ffnumber']; ?></td>
                        <td><?php echo $l['fname']; ?></td>
                    </tr>
                    <tr>
                        <td>Mother </td>
                        <td><?php echo $l['mfnumber']; ?></td>
                        <td><?php echo $l['mname']; ?></td>
                    </tr>
                    <tr>
                        <td>Location </td>
                        <td><?php echo $l['location']; ?></td>
                    </tr>
                    <tr>
                        <td>Owner </td>
                        <td><?php echo $l['care']; ?></td>
                    </tr>
                    <!-- marks the line between horses -->

    <tr>/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
\/\/\/\/\/\/\/\/\/\</tr>
                    <tr> </tr>
                </tr>
            </table>

<?php

    }
}

echo "<br><br>";

?>


</body>
</html>
```

14. wordi.php

```
<?php
include 'db.inc.php';


/*
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.05.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: wordi.php
		Connected files: mainReport.php (restriction sent), db.inc.php (the connection to the database)
		Function: to show a list of all horses depending on restriction chosen by the user
		Output: a word document

*/

// the type we are outputting
header("Content-Type: application/msword");

//$fieldChosen = '';
//gets the data chosen in the drop down list and determines if the variable is an array
if(is_array($_POST['fields'])) {
		//looping through the values in the array
		foreach($_POST['fields'] as $fieldChosen) {
			//printing out the name of the colour
			$fields = $fieldChosen;
		}
} else {
		echo 'No restriction was selected';

}

//the query is sent to the database
$result = mysql_query("SELECT * FROM hestur ORDER BY $fields");
// if the query is not good
if(!$result) {
		// print an erro message
		exit('Database error: ' . mysql_error());
}

// check if the number of rows is null
if(mysql_num_rows($result) > 0) {
//get the data from the array
		while($l = mysql_fetch_array($result)) {
			//print out the data
			echo "<table border=\"1\">";
			echo "<tr>";
				echo "<tr>";
					echo "<td><b><u>Nr. ". $l['id']."</u></b></td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					echo "<td>Registration date: ". $l['day']."</td>";
				echo "</tr>";
				echo "<tr><td> </td></tr>";
				echo "<tr>";
					echo "<td>ID Number: </td>";
					echo "<td>". $l['fnumber']."</td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
```

```
                                        echo "<td>  </td>";
                                        // formating a number with grouped thousands
                                        echo "<td>Price: <u>". number_format($l['price'], 2, ',', ' ')."</u>
kr.</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Name: </td>";
                                        echo "<td>". $l['name']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Colour: </td>";
                                        echo "<td>". $l['litirid']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>----------</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Father </td>";
                                        echo "<td>". $l['ffnumber']."</td>";
                                        echo "<td>  </td>";
                                        echo "<td>Name: </td>";
                                        echo "<td>". $l['fname']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Mother </td>";
                                        echo "<td>". $l['mfnumber']."</td>";
                                        echo "<td>  </td>";
                                        echo "<td>Name: </td>";
                                        echo "<td>". $l['mname']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Location </td>";
                                        echo "<td>". $l['location']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Owner </td>";
                                        echo "<td>". $l['care']."</td>";
                                echo "</tr>";
                                /*<!-- marks the line between horses -->*/
                                echo "<tr> </tr>";
                        echo "</tr>";
                        echo "</table><br /><br />";
        }
} else {
        // else leave a message that there are no horses in the database
        exit('There are no horses in the database.');
}


?>
```

15. excel.php

```
<?php
include 'db.inc.php';

/*
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.05.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: excel.php
		Connected files: mainReport.php (restriction sent), db.inc.php (the connection to the database)
		Function: to show a list of all horses depending on restriction chosen by the user
		Output: an excel spreadsheet

*/

// the type we are outputting
header("Content-Type: application/vnd.ms-excel");
//$fieldChosen = '';
//gets the data chosen in the drop down list and determines if the variable is an array
if(is_array($_POST['fields'])) {
		//looping through the values in the array
		foreach($_POST['fields'] as $fieldChosen) {
			//insert the data into a variable
			$fields = $fieldChosen;
		}
} else {
		// print out a message if there is no restriction selected
		echo 'No restriction was selected';

}

//the query is sent to the database
$result = mysql_query("SELECT * FROM hestur ORDER BY $fields");
// if the query is not good
if(!$result) {
		// print an erro message
		exit('Database error: ' . mysql_error());
}

//check if the number of rows is null
if(mysql_num_rows($result) > 0) {
//get the data from the array
		while($l = mysql_fetch_array($result)) {
			//print out the data
			echo "<table border=\"1\">";
			echo "<tr>";
				echo "<tr>";
					echo "<td><b><u>Nr. ". $l['id']."</u></b></td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					echo "<td>Registration date: ". $l['day']."</td>";
				echo "</tr>";
				echo "<tr><td> </td></tr>";
				echo "<tr>";
					echo "<td>ID Number: </td>";
					echo "<td>". $l['fnumber']."</td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					echo "<td>  </td>";
					// formating a number with grouped thousands
					echo "<td>Price: <u>". number_format($l['price'], 2, ',', ' ')."</u> kr.</td>";
				echo "</tr>";
				echo "<tr>";
					echo "<td>Name: </td>";
					echo "<td>". $l['name']."</td>";
				echo "</tr>";
				echo "<tr>";
```

```
                                        echo "<td>Colour: </td>";
                                        echo "<td>". $l['litirid']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>----------</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Father </td>";
                                        echo "<td>". $l['ffnumber']."</td>";
                                        echo "<td>  </td>";
                                        echo "<td>Name: </td>";
                                        echo "<td>". $l['fname']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Mother </td>";
                                        echo "<td>". $l['mfnumber']."</td>";
                                        echo "<td>  </td>";
                                        echo "<td>Name: </td>";
                                        echo "<td>". $l['mname']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Location </td>";
                                        echo "<td>". $l['location']."</td>";
                                echo "</tr>";
                                echo "<tr>";
                                        echo "<td>Owner </td>";
                                        echo "<td>". $l['care']."</td>";
                                echo "</tr>";
                                /*<!-- marks the line between horses -->*/
                                echo "<tr> </tr>";
                        echo "</tr>";
                        echo "</table><br /><br />";
        }
} else {
        // else leave the message that there are no horses in the database
        exit('There are no horses in the database.');
}


?>
```

16. search.php

```
<?php
        include 'main.php';
        include 'db.inc.php';
?>
<!--
        Author: Ólafur Hannesson
        Date: (final changes and commenting) 05.04.2008
        Project: part of final year project autumn semester 2007, spring semester 2008
        Name of file: search.php
        Connected files: result.php (the results displayed), main.php (the outlook of the page and menu), footer.php (the
footer of the page)
                                                db.inc.php (the connection to the database)
        Function: Search for data in database and upon submitting return the results in separate file, result.php
-->

<!--
        this file is the search form, it allows the user to pick from some drop down list
        where he wants to get the data from. once the user hits the 'Go!!' button, a new window will open,
        which will display the data searched for or left blank, display all data.
-->
                        <!-- this is the form the user will have to pick a field to search in and put in  -->
                        <!-- the data to search for
                                                                                -->
                        <p><h1>Searching.....</h1></p>
                        <ul>
                        <form method="post" action="result.php" target="_blank">
                                <table border="0" cellpadding="0" cellspacing="0">
                                <tr>
                                        <td>
                                                <p align="center">
                                                <!-- the fields to search in -->
                                                <select name="fields" size="1">
                                                <option value="fnumber">ID Number</option>
                                                <option value="name">Name</option>
                                                <option value="ffnumber">Fathers ID Number</option>
                                                <option value="fname">Fathers name</option>
                                                <option value="mfnumber">Mothers ID Number</option>
                                                <option value="mname">Mothers name</option>
                                                <option value="location">Location</option>
                                                </select>
                                                <!-- the text box the user must fill in to search for -->
                                                <input type="text" name="search" size="25">  <br>
                                                <br />
                                                Search database: <input type="submit" value="Search"
name="Go"></p>
                                        </td>
                                </tr>
                                </table>
                        </form>
                        </ul>
                        <p><br /></p>


<?php include 'footer.php' ?>
```

17. result.php

```
<?php
		include 'main.php';
		include 'db.inc.php';


?>
<!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: result.php
		Connected files: search.php (the searching form), main.php (the outlook of the page and menu), footer.php (the
footer of the page)
                                                db.inc.php (the connection to the database)
		Function: find and print out the results from the search from search.php
-->

<!--
		this file gets the parameters from search.php, goes and compares all the data to the information given and prints out
the results.
		the input needs to be as precise as possible, since 'Alice' will be found from 'Alice' but not from 'Alice in Wonderland'
-->

		<!-- the table the results appear in -->
  <table border="1" cellpadding="5" cellspacing="0">
  <tr>
  <!-- the column headers -->
  <td width="100"><b>ID</b></td>
  <td width="90"><b>ID number</b></td>
  <td width="100"><b>Name</b></td>
  <td width="90"><b>Fathers ID number</b></td>
  <td width="100"><b>Fathers Name</b></td>
  <td width="90"><b>Mothers ID number</b></td>
  <td width="100"><b>Mothers Name</b></td>
  <td width="100"><b>Location</b></td>
  </tr>
  <tr>
  <td>
  <?php
  //gets the data to search for
		$search = $_POST['search'];
		//gets which column to look in
		$fields = $_POST['fields'];
  //the query is sent to the database
  $result = mysql_query("SELECT * FROM hestur WHERE $fields LIKE '%$search%' LIMIT 0, 250");
		//query details table begins
  while ($row = @mysql_fetch_array($result)) {
		//gets the data from each row
		  $id=$row["id"];
		  $fnumber=$row["fnumber"];
		  $name=$row["name"];
		  $ffnumber=$row["ffnumber"];
		  $fname=$row["fname"];
		  $mfnumber=$row["mfnumber"];
		  $mname=$row["mname"];
		  $location=$row["location"];

		  //table layout for results
		  echo ("<tr>");
		  echo ("<td>$id</td>");
```

```
                echo ("<td>$fnumber</td>");
                echo ("<td>$name</td>");
                echo ("<td>$ffnumber</td>");
                echo ("<td>$fname</td>");
                echo ("<td>$mfnumber</td>");
                echo ("<td>$mname</td>");
                echo ("<td>$location</td>");
                echo ("</tr>");
    }

    //these are the results if there is no data found
    if (!$id) {
            echo "</table>";
            echo "No Record Found, to search again please close this window";
    }

    ?>
    </table>


<?php include 'footer.php' ?>
```

18. check.php

```
<?php
	include 'main.php';
	include 'db.inc.php';
?>
<!--
	Author: Ólafur Hannesson
	Date: (final changes and commenting) 05.04.2008
	Project: part of final year project autumn semester 2007, spring semester 2008
	Name of file: check.php
	Connected files: hestur.php (the results from the registration form), main.php (the outlook of the page and menu),
				footer.php (the footer of the page),
				db.inc.php (the connection to the database),
				reports.php (the complete list of horses in database)
	Function: to print out all the data in the database
-->

<!--
	the file lists all the horses in the database and allows the user to delete from the database.

-->

<?php
//defines the variable 'action' and gets the data
if(isset($_GET['action'])) {
	$action = $_GET['action'];
} else {
	$action = '';
}

if($action == 'edit' and isset($_GET['fnumber'])) {
	//if action is 'edit' and the fnumber is set
	if(isset($_POST['fnumber'])) {
		//if the variable is set, get the data from text fields
		$location = $_POST['location'];
		$notes = $_POST['notes'];
		$fnumber = $_POST['fnumber'];

		//create a query
		$sqlUP = mysql_query("UPDATE 'hestur' SET 'location' = '$location', 'notes' = '$notes' WHERE 'fnumber' =
'$fnumber' LIMIT 1");
		//if the query is good
		if($sqlUP) {
			//print successful message
			echo 'horse details have been updated! ';
			echo '<a href="check.php">';
		} else {
			//else print error message
			echo '<p>Error updating horse details: ' . mysql_error() . '</p>';
		}


	} else {
		//if the variable is not set, get the fnumber
		$fnumber = $_GET['fnumber'];
		//create and send the query
		$list = mysql_query("SELECT id, fnumber, name, ffnumber, fname, mfnumber, fname, location, notes
FROM hestur WHERE fnumber='$fnumber'");
		//if the query is not good...
		if (!$list) {
			//print an error message and terminate the script
```

```
                exit('<p>Error fetching horse details: ' . mysql_error() . '</p>');
            }

            //get the result row as an array
            $horseList = mysql_fetch_array($list);

        ?>
            <!-- print out the form for editiing, the id, fnumber and name are not changeable -->
            <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
                    <table cellspacing="10px">
                <tr>
                  <td>id
                  <td><?php echo $horseList['id']; ?>
                </tr>
                                <tr>
                  <td>ID number:
                  <td><?php echo $horseList['fnumber']; ?>
                </tr>
                <tr>
                  <td>Name:
                  <td><?php echo $horseList['name']; ?>
                </tr>
                                <tr>
                  <td>Location:
                  <td><input type="text" name="location" value="<?php echo $horseList['location']; ?>" />
                </tr>
                <tr>
                  <td>Notes:
                  <td><input type="text" name="notes" value="<?php echo $horseList['notes']; ?>" />
                </tr>
              </table>
                    <input type="hidden" name="fnumber" value="<?php echo $fnumber; ?>" />
                    <input type="submit" value="Submit" name="submitEdit" />
            </form>

        <?php
        }

} elseif($action == 'view' and isset($_GET['fnumber'])) {
        // if the action is 'view', and the fnumber is set
        // get the fnumber
        $id = $_GET['fnumber'];
        // create the table
        ?>
        <table border="1" cellpadding="5" cellspacing="0">
        <?php
        // create the query
        $sqlCheck = "SELECT * FROM hestur WHERE fnumber = '$id'";
        //send the query
        $list = @mysql_query($sqlCheck);
        //if the results are not good
        if(!$list) {
                //print an error message
                exit('Database error: ' . mysql_error());
        }

        // if the results are good, check if the number of rows in the result is not empty
        if(mysql_num_rows($list) > 0) {
                // get the data from the array and put it in a variable
                while($l = mysql_fetch_array($list)) {
                        //print the results
                        ?>
```

```
                    <tr>
                        <td>Registration day</td>
                        <td><?php echo $l['day']; ?></td>
                        <td>Price</td>
                        <td><?php echo $l['price']; ?></td>
                    </tr>
                    <tr>
                        <td>ID Number</td>
                        <td><?php echo $l['fnumber']; ?></td>
                    </tr>
                    <tr>
                        <td>Name</td>
                        <td><?php echo $l['name']; ?></td>
                    </tr>
                    <tr>
                        <td>Colour</td>
                        <td><?php echo $l['litirid']; ?></td>
                    </tr>
                    <tr>
                        <td>Fathers ID number:</td>
                        <td><?php echo $l['ffnumber']; ?></td>
                    </tr>
                    <tr>
                        <td>Fathers name:</td>
                        <td><?php echo $l['fname']; ?></td>
                    </tr>
                    <tr>
                        <td>Mothers ID number:</td>
                        <td><?php echo $l['mfnumber']; ?></td>
                    </tr>
                    <tr>
                        <td>Mothers name</td>
                        <td><?php echo $l['mname']; ?></td>
                    </tr>
                    <tr>
                        <td>Owner</td>
                        <td><?php echo $l['care']; ?></td>
                    </tr>
                    <tr>
                  <td>Location:
                  <td><?php echo $_POST['location']; ?>
                 </tr>
                 <tr>
                  <td>Notes:
                  <td><?php echo $_POST['notes']; ?>
                 </tr>

            <?php
                        }
            }
                        ?>
            </table>
            <?php
            echo '<br /><br />';
            // the user can click the link when he has viewed the data
            exit('Please click <a href="check.php">here</a> to return.<br /><br />');

} elseif($action == 'del' and isset($_GET['fnumber'])) {
            //if the user wants to delete the horse the action is 'del' and fnumber is set
            //get the fnumber
            $id = $_GET['fnumber'];
```

```
		//create the query
		$sql = "DELETE FROM hestur WHERE fnumber = '$id'";
		//check the results
		if(mysql_query($sql)) {
			//if the results are good
			echo '<p>Your horse has been successfully deleted from the database!</p>';
			echo '<p>Press <a href="check.php">here</a> to return!</p>';
		} else {
			// if the results are bad
			echo '<p>There is a problem: ' . mysql_error() . '</p>';
		}

//		header('location: ' . $_SERVER['PHP_SELF']);
// has printed out all relevant messages and terminates the script
		exit();
}
?>
<!-- the form for the page -->
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<p><h2>The following horses are registered in the database: </h2></p>
<!-- the table for the data -->
<table border="1" cellpadding="5" cellspacing="0">
<tr>
		<!-- column headers -->
		<th>  </th>
		<th> Date </th>
		<td></td>
		<th> ID number</th>
		<th> Name </th>
		<th> Location </th>
		<th></th>
		<th></th>
</tr>
<?php
// get all the data from table 'hestur', query created
$sqlh = "SELECT * FROM hestur";
//send the query
$list = mysql_query($sqlh);
//if the results are not good
if(!$list) {
		exit('Database error: ' . mysql_error());
}

// if the number of rows is zero
if(mysql_num_rows($list) > 0) {
		//get the data in the array into a variable
		while($l = mysql_fetch_array($list)) {
			//print out the results
			?>
		<tr valign="top">
			<td></td>
			<td><?php echo $l['day']; ?></td>
			<td><?php echo $l['id']; ?>.</td>
			<!-- if the user wants to view the information about the horse -->
			<td><a href="<?php echo $_SERVER['PHP_SELF']; ?>?action=view&fnumber=<?php echo $l['fnumber'];
?>"><?php echo $l['fnumber']; ?></a></td>
			<td><?php echo $l['name']; ?></td>
			<td><?php echo $l['location']; ?></td>
			<!-- if the user wants to edit the information about the horse or delete the horse -->
			<td>[ Edit |
		<a href="<?php echo $_SERVER['PHP_SELF']; ?>?action=del&fnumber=<?php echo $l['fnumber']; ?>" onclick="return
confirm('Delete this horse from the database?');">Delete</a> ]</td>
```

```
            </tr>
<?php
            }
} else {
            // if there are no horses in the database
            ?>
            <tr><td colspan="3">No horses</td></tr>
            <?php
}
?>
</form>
</table>
<br />
<p> </p>


<?php include 'footer.php' ?>
```

19. photo.php

```
<?php
		include 'main.php';
		include 'db.inc.php';
?>
<!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: photo.php
		Connected files: main.php (the outlook of the page and menu), footer.php (the footer of the page)
                                                db.inc.php (the connection to the database)
		Function: to keep up to data a list of photos and information
-->

<!--
		this file allows the user to put and maintain a image gallery. the user can delete from the list or
		view the pictures
-->
<?php
//an action declaration created and determined

if (isset($_GET['action'])) {
  $action = $_GET['action'];
} else {
  $action = '';
}

if ($action == 'del' and isset($_GET['id'])) {
  $id = $_GET['id'];

  // User is deleting a file
  $sql = "DELETE FROM filestore WHERE id = '$id'";
  $ok = mysql_query($sql);
  if (!$ok) {
    exit('Database error: ' . mysql_error());
  }else { // print successful messages
		exit('Congratulations, you have deleted a photo. Please click <a href="photo.php">here</a> to return.');
		  } //print relevant messages and terminate the script
  exit();

} elseif (isset($_FILES['upload'])) {
		//insert data into variables
		$uploadfile = $_FILES['upload']['tmp_name'];
		$uploadname = $_FILES['upload']['name'];
		$uploadtype = $_FILES['upload']['type'];
		$uploaddesc = $_POST['desc'];

		// Open file for binary reading ('rb')
		$tempfile = fopen($uploadfile, 'rb');

		// Read the entire file into memory using PHP's
		// filesize function to get the file size.
		$filedata = fread($tempfile, filesize($uploadfile));

		// Prepare for database insert by adding backslashes
		// before special characters.
		$filedata = addslashes($filedata);

		// Create the SQL query.
		$sql = "INSERT INTO filestore SET
		  filename = '$uploadname',
		  mimetype = '$uploadtype',
		  description = '$uploaddesc',
		  filedata = '$filedata'";

		// Perform the insert.
		$ok = mysql_query($sql);
```

```
		if ($ok) {
			echo 'You have inserted a photo into the database. Please click <a href="photo.php">here</a> to return.';
			exit();
		} else {
			exit('Database error storing file: ' . mysql_error());
		}
		exit();
}

// Default page view: lists stored files
// get all the data from table 'filestore'
$sql = 'SELECT id, filename, mimetype, description FROM filestore';
//send the query
$filelist = mysql_query($sql);
// if the results are not good
if (!$filelist) {
		//print error message and terminate the script
  exit('Database error: ' . mysql_error());
}
?>

<!-- the upload form -->
<form action="<?php echo $_SERVER['PHP_SELF']; ?>"
   method="POST" enctype="multipart/form-data">
   <table>
   <tr><td></td></tr>
<tr>
		<td>Insert image: </td>
		<td><input type="file" name="upload" /></td>
</tr>
<tr>
		<td>Description:</td>
		<td><input type="text" name="desc" maxlength="255" /></td>
</tr>
<tr><td><input type="submit" value="Upload" /></td></tr>

</table>
</form>

<!-- the table to list all the pictures -->
<p>The following files are stored in the database:</p>
<table border="1" cellpadding="5" cellspacing="0">
<tr>
  <th>ID </th>
  <th>Filename</th>
  <th>Type</th>
  <th>Description</th>
</tr>
<?php

//check if the number of rows is empty
if (mysql_num_rows($filelist) > 0) {
		// while the rows are not empty, get them into variable
  while ($f = mysql_fetch_array($filelist)) {
		//print them out in the table
    ?>

<tr valign="top">
  <td><?php echo $f['id']; ?></td>
  <td>
    <a href="photoS.php?action=view&id=<?php echo $f['id']; ?>">
      <?php echo $f['filename']; ?></a>
  </td>
  <td><?php echo $f['mimetype']; ?></td>
  <td><?php echo $f['description']; ?></td>
  <td>
    [ <a href="<?php echo $_SERVER['PHP_SELF']; ?>?action=del&id=<?php echo $f['id']; ?>"
      onclick="return confirm('Delete this file?');"> Delete </a>]
  </td>
</tr>
```

```
   <?php
  }
} else {
            //or if the number of rows is empty, tell the user that it is empty
  ?>
  <tr><td colspan="3">No Files!</td></tr>
  <?php
}
?>
</table>


<?php
            include 'footer.php';
?>
```

20. photoS.php

```
<?php

include 'db.inc.php';

if (isset($_GET['action'])) {
 $action = $_GET['action'];
} else {
 $action = '';
}

if (($action == 'view') and isset($_GET['id'])) {
 $id = $_GET['id'];

 // User is retrieving a file
 $sql = "SELECT filename, mimetype, filedata, description
    FROM filestore WHERE id = '$id'";
 $result = @mysql_query($sql);
 if (!$result) {
  exit('Database error: ' . mysql_error());
 }

 $file = mysql_fetch_array($result);
 if (!$file) {
  exit('File with given ID not found in database!');
 }

 $filename = $file['filename'];
 $mimetype = $file['mimetype'];
 $filedata = $file['filedata'];
 $descript = $file['description'];
 $disposition = 'inline';

 header("content-disposition: $disposition; filename=$filename");
 header("content-type: $mimetype");
 header('content-length: ' . strlen($filedata));

 echo($filedata);

 exit();

} else {
        exit('File with given ID not found in database!');
}
?>
</table>
</body>
</html>
```

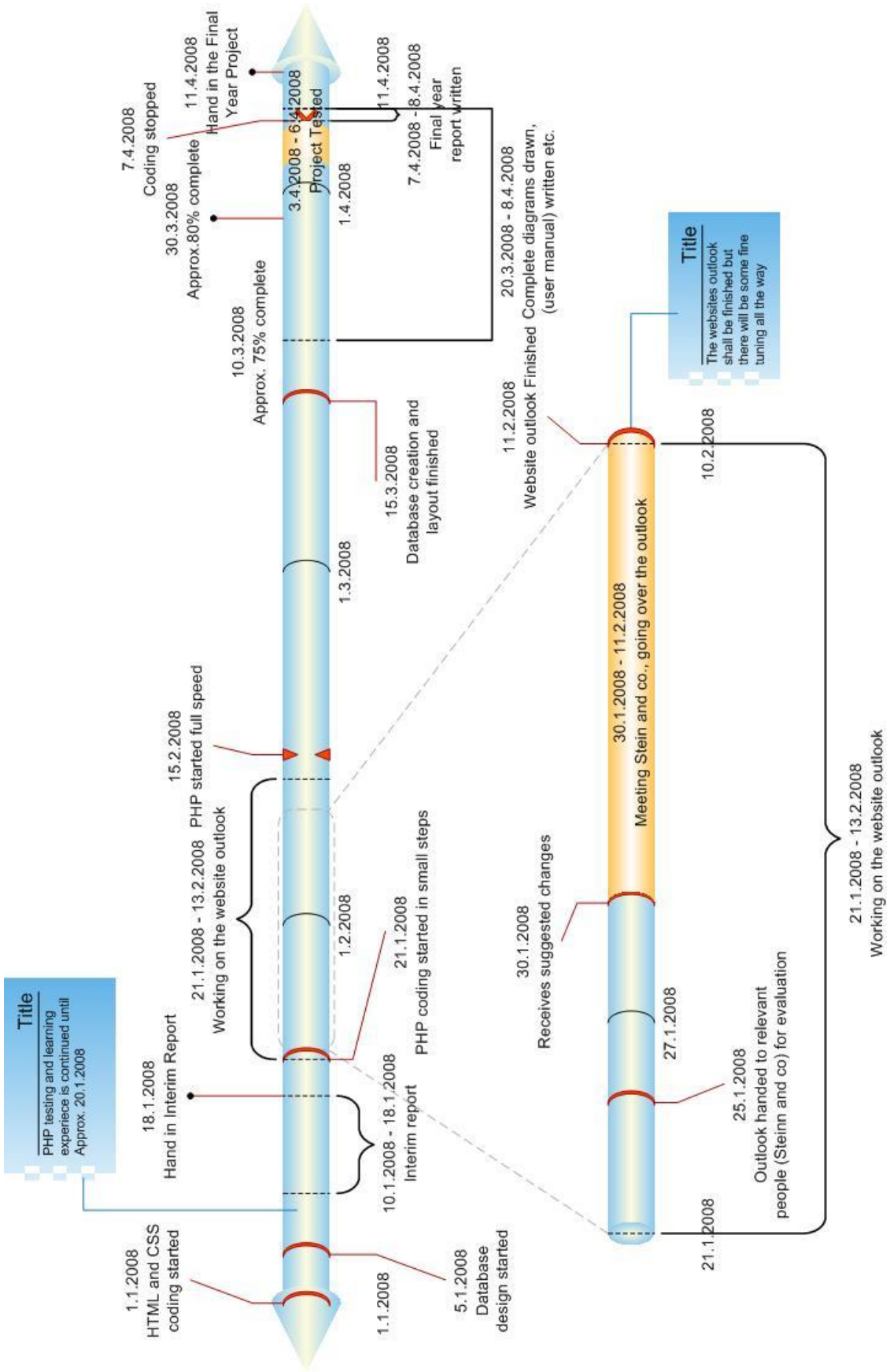
21. about.php

```
<?php include 'main.php';?>
 <!--
		Author: Ólafur Hannesson
		Date: (final changes and commenting) 05.04.2008
		Project: part of final year project autumn semester 2007, spring semester 2008
		Name of file: about.php
		Connected files: main.php (the outlook of the page and menu), footer.php (the footer of the page)
		Function: about the project
-->
		<p><h1><center>...About...</center></h1></p>
			<p>
			This is a final year project done in the 2007-2008 semesters.
			The project was actually started 2-3 years ago when I was asked to create a database with an interface.
<br />
			Microsoft Access was chosen and the database was made. That database did not work out as it should be
and was abandoned later.

			Autumn 2007, I was looking for a project to do as my final year project in Computer Science at the
University of Akureyri. One project idea from Nik Whitehead, my supervisor,
			looked to me to be almost the same as the database idea I had worked at in Access. This time I worked
with the help of PHP, MySQL and Apache and created a web page interface
			with MySQL database underneath.
			<ol type="1" start="1">
				<li>1. The version of PHP or <i>PHP: Hypertext Processor</i> I used is 5.2.4.</li>
				<li>2. The version of MySQL I used is 5.0.45.</li>
				<li>3. The version of Apache I used is 2.0.59.</li>
				<li></li>
			</ol>
			<br />
			<p>The interface is designed in:</p>
			<ol type="1" start="1">
				<li>1. phpDesigner2008 version 6.0.1.2. gotten from <a href="http://www.mpsoftware.dk/"
target="_blank">MPSOFTWARE</a></li>
				<li>2. Notepad version 5.1 from Microsoft (already in Windows)</li>
				<li>3. Notepad++ version 4.6, open source software, freely distributed on sourceforge.net</li>
				<li><br /><br /></li>
				<li>4. Mozilla Firefox version 2.0.0.13 freely distributed on the <a
href="http://www.mozilla.com" target="_blank">Mozilla homepage</a></li>
				<li>5. Internet Explorer version 7.0.5730.13 (already installed in Windows)</li>
			</ol>
			<br />
			The project was finished by 9th of April and handed in the following day, 10th of April.
			<br /><br /><br /><br /><br /><br />
			<u>Contact information</u>:<br />
			<b><i>Ólafur Hannesson</i></b><br />
			email: <a href="mailto:olafurhannesson@gmail.com">olafurhannesson@gmail.com</a><br />
			</p>


<?php include 'footer.php' ?>
```

Appendix B: Complete Project plan



Appendix C: Sample data

1 - Hestar

Id	1.
Day	15.04.2008
ID Number	**IS2002186810**
Name	Gæi
ID number father	IS1986186055
Name of father	Orri
ID number mother	IS1990286807
Name of mother	Álfheiður Björk
Litirid	1
Care	Þórunn Guðlaugsdóttir
Location	Lækjarbotnum
Notes	NULL
Price	150000

(1) An example here would be 1 which would be picture with id 1 in table Filestore

2 - Litir

id	litur	Description
1.	Grár/Jarpur	NULL
2.	Bleikur	Kind of white

3 - Personnel

id	name	address	phone	gsm	notes
1.	Þórunn Guðlaugsdóttir	Lækjarbotnum	NULL (1)	NULL (2)	NULL

(1) Here an example would be 487-5489

(2) Here an example would be 868-4022

4 - Father

Id	1.
ID number	IS1986186055
Name	Orri
Gender	Male
Colour	Unkown
ID number father	IS1982151001
Name of father	Otur
ID Number mother	IS1983284555
Name of mother	Dama
Ownerid	Unknown
phone	NULL
Location	Þufa

5 - Mother

Id	1.
ID Number	IS1990286807
Gender	Female
Name	Álfheiður Björk
Colour	Bleikur
ID Number father	IS1986165509
Name of father	Bliki
ID Number mother	IS1983287806
Name of mother	Hekla Mjöll
Ownerid	Unknown
phone	NULL
Location	Lækjarbotnum

Appendix D: User Manual

1. About project

This is a final year project, done at the University of Akureyri with the supervision of Nik Whitehead. The final year project is a project that stretches over two semesters in the year 2007 to 2008, (autumn and spring).

This project has between 2 and 3 year prelude where it was started by some interface creation in Microsoft Access and some database creation by using some templates. That program was later deemed not suitable to use as it did not fulfil completely what was pursued. That program was also a bit heavy in use and while some modifications were done, using Visual Basic coding, it was discarded shortly after its creation.

One proposal from Nik Whitehead was very similar to this project, although that project may have had bit more data involved. There is enough data involved in this project and the main matter is to be able to put the data in, organize it and create some satisfactory outputs.

The installation process is described in chapter 2. Chapter 3 covers the programming language and the database system used along with the server. “Other things” is about the system requirements. Chapter 4 to 7 describe and give some samples for the functions in the interface. Chapter 8 is about the database. Appendices A through C are about the installation of the actual packages should the user want to install them onto the computer instead of running it of a USB stick.

2. Installation

The best way to work on this webpage is to install the Apache, PHP and MySQL, and phpMyAdmin packages as is shown in appendices A – D, onto the computer it is intended to work on. After the installations are finished, go to the phpMyAdmin and import the file db.sql that is on the following CD. After that any registration can be concluded.

The only thing the user needs to know is if any of the packages are already installed on the computer. It is recommended to shut them down before starting up WOS. Should the computer be connected to the internet, it is recommended to disconnect before starting up WOS. After the user has plugged the stick in, he needs to double click on the WOS icon to start the web server. After that the user can start registering horses.

3. Tools

    a. Other things

The project is created using (on two computers):

- Microsoft Windows® XP/XP2
- (Computer 1) 3.50 GB RAM (Computer 2) 356 MB RAM
- Using <20 MB Hard Disk Space
- Mozilla Firefox 2.0.0.13 and Internet Explorer 7.0.5730.13
- phpDesigner2008 version 6.0.1.2 from MPSoftware
- Notepad version 5.1 from Microsoft
- Notepad++ version 4.6, distributed from sourceforge.net

There are really no known system requirements for this project package (after testing). However it is recommended that the user uses versions close to,

- Mozilla Firefox 1.9+ and Internet Explorer 4+

and has at least 20 MB space free on the hard disk.

b. PHP

PHP or PHP: Hypertext Preprocessor is a programming language that was designed for producing dynamic web pages. It is still a popular misconception that PHP stands for Personal Home Page Tools as its name was in the beginning. According to the official PHP webpage, PHP is "*a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML … the main goal of the language is to allow web developers to write dynamically generated web pages quickly." [* (PHP: Hypertext Preprocessor, 2007)*]* PHP is easy to use when the intent is to create simpler websites from some very complicated and vast amount of data. Its most popular choice is to use in server side scripting but is also used in graphical applications or from a command line interface. PHP is open source server-side language that is very easy to learn not to mention very powerful. PHP is becoming very popular choice along with MySQL and Apache to create database driven web pages. The reason for its popularity might be very simple. PHP uses simple syntax, i.e. the scripts are kept as readable as humanly possible which in turn allows for flexibility. This flexibility however leads to security concerns as well as performance issues. Rasmus Lerdorf is titled the father of PHP when he started with it in 1994. He used if mainly for maintaining his personal home page. PHP2 came out year later and PHP3 in 1997. Today's version is number 5.2.5 and was released 8. November 2007. I downloaded that version and set up but the version I had before was number 5.2.3.

c. MySQL

MySQL is a high performance multi-user relational database management system that today is the most popular choice when it comes to database driven web pages. The reason for its popularity is preferably the easy way it has when it comes to maintaining it. MySQL is open source SQL database system that has been developed, supported and distributed by MySQL AB which was founded by the MySQL developers. The current stable version of MySQL is 5.1.22 but the newest version is 6.0.3.alpha, released unknown. I did get 5.0.45 since I had some major problems concerning communications between the webpage and the database. Other problems I encountered were for example problems handling strings and numbers, issues regarding privileges and connection problems from the website into the database. I have used phpMyAdmin occasionally to help with the database creation and maintenance as well as making sure that everything is as it should be. phpMyAdmin is basically a help program that "*can manage a whole MySQL server (needs a super-user) as well as a single database."* [(Ratschiller & al, 2007)]

d. Apache

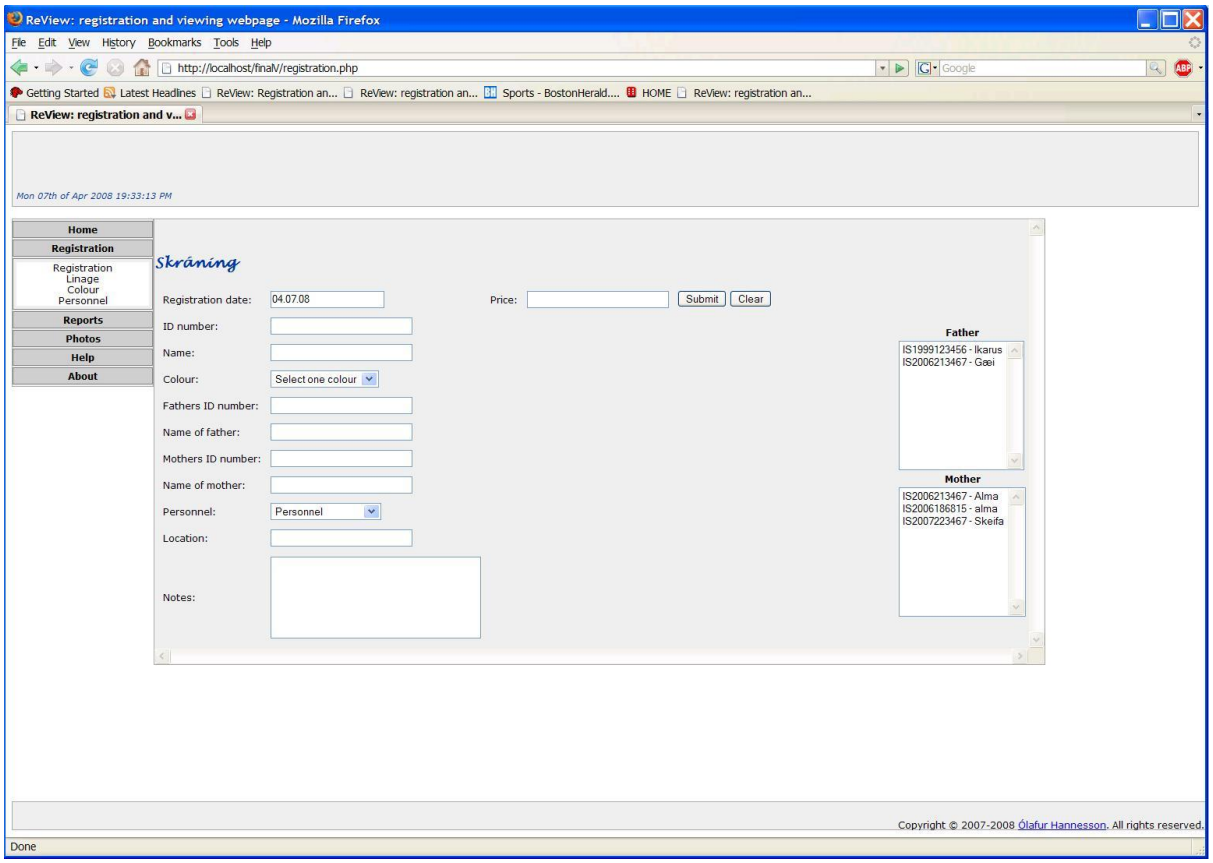
What is Apache? According to the Apache website, "*The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful and freely-available source code implementation of an HTTP (Web) server."* [(Behlendorf & al, 2002)] Apache HTTP Server or more commonly Apache is the market leading free web server (according to NetCraft [(NetCraft, 2007)]). While Apache is developed and supported by the open source

community, it is freely distributed and is easy to set up. Many website designers have installed and set up Apache on their machines to enable easy way to preview and test websites while being developed. Apache was first created by Robert McCool sometime in the early 1990s with its next version released in 2002. Today's version is 2.2.6, released 7. September 2007. I downloaded Apache version 2.0.59. I did not go with earlier version for the simple reason that there were some compatibility issues between Apache, MySQL and PHP (the newest versions available ate the time). Today, 12. January there is still some issues and it is recommended to use some older versions.

e. HTML and CSS

Currently most websites that employ HTML employ CSS as well. HTML stands for Hypertext Mark-up Language and enables us to mark the text so it can function as hypertext on the web. Editors like Notepad++ and Microsoft FrontPage, mark the manuscript pages with symbols that tells the web how to print the page. HTML uses its own set of symbols on how the browsers should display the pages. CSS or Cascading Style Sheet was designed to help the designers with the formatting of the web pages, comparable to the styles in programs like Microsoft Word. CSS provides a way for the designer to specify the formatting in one file and have the changes reflected all over the website instead of having to lookup the formatting on many web pages. By using CSS the designer is not only guaranteeing himself a way of making the life of maintaining the website much easier, but it will also make it so that when a new person comes along, that person can quickly set herself into the outlook of the website.

4. Registration



a. Registration

Picture 6: Registration

The registration process is fairly simple. As is shown in Picture 1 the *Registration* is under the Registration label, along with *Linage, Colour and Personnel*. *Registration* asks the user to enter various information including the ID number, name parentage, owner and location.

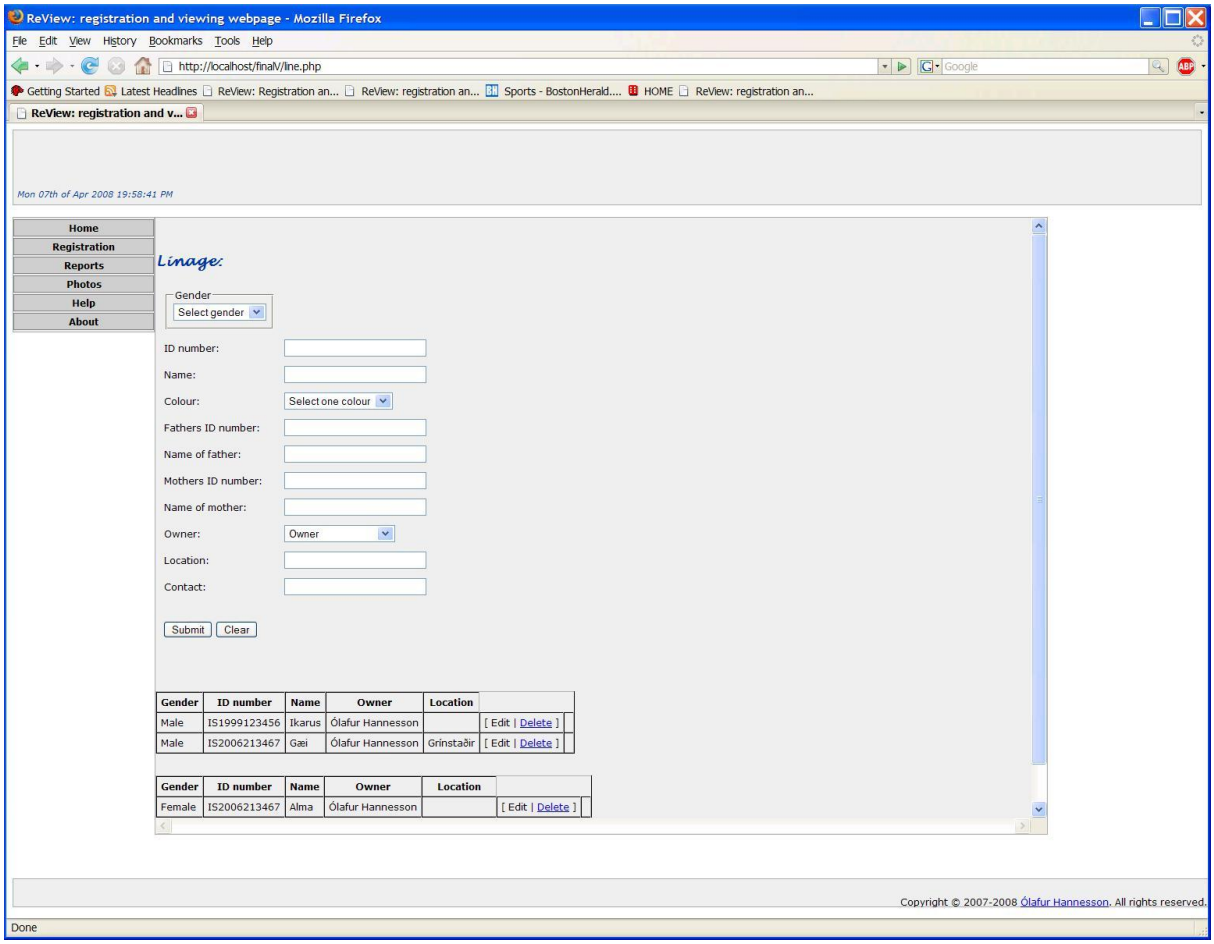
- The **ID number** is a 12 digit string, e.g. IS2008286801. This must be filled in for the registration to be successfully concluded.
- The **Name** is the name of the horse,
- The **Colour** is the just that, the colour of the horse. The user must have created the colour before.
- The **Fathers ID number** is the id number as is with the ID number mentioned before and shown in the table at the side, along with the **Name of father** which is the name of the horse father. These information can be seen in a table to the side provided that the horse has been registered before as potential parent
- The **Mothers ID number** is the id number as is with the ID number mentioned before and shown in the table at the side, along with the **Name**

of mother which is the name of the horse mother. These information can be seen in a table to the side provided that the horse has been registered before as potential parent

- **Personnel** is the owner of the horse
- **Location** is where the horse is situated,
- **Notes** can be any text that is 255 characters long. This can be any information about the horse, i.e. owner history, other shows he has been at etc.
- **Price** is the rough price the owner puts on the horse

The tables at the side get their information from the *Linage* registration form and hold information about potential parents.

b. Linage



Picture 7: Linage

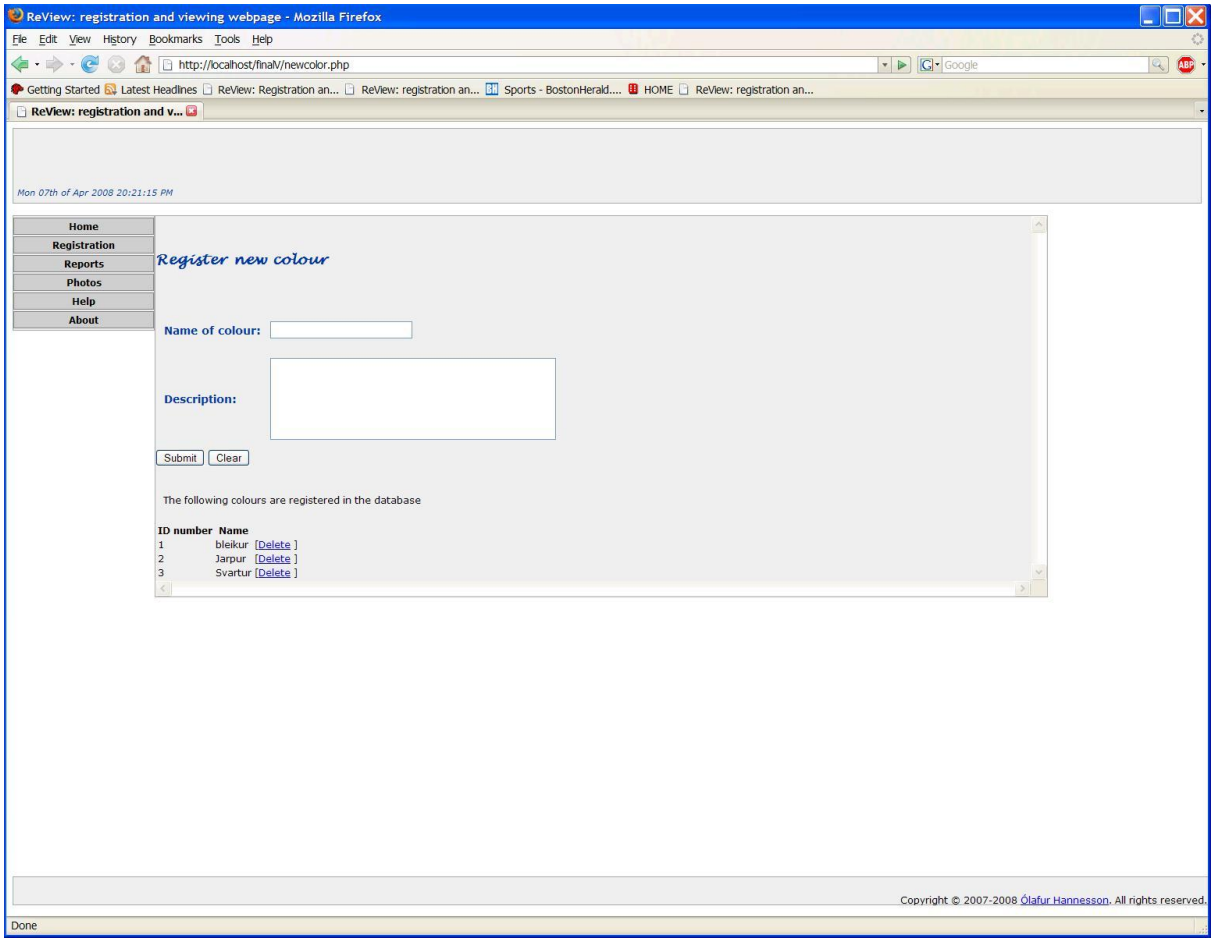
Linage registration form (see Picture 2) is very simple and holds most of the same text boxes as the registration form.

- The **Gender** is defined in the drop down list. This is to prevent any confusion when the user is writing in the data about the horse.
- The **ID number** is a 12 digit string, e.g. IS2008286801. This must be filled in for the registration to be successfully concluded.
- The **Name** is the name of the horse,
- The **Colour** is the just that, the colour of the horse. The user must have created the colour before.
- The **Fathers ID number** is the id number as is with the ID number mentioned before and shown in the table at the side, along with the **Name of father** which is the name of the horse father. These information can be seen in a table to the side provided that the horse has been registered before as potential parent

- The **Mothers ID number** is the id number as is with the ID number mentioned before and shown in the table at the side, along with the **Name of mother** which is the name of the horse mother. These information can be seen in a table to the side provided that the horse has been registered before as potential parent
- **Owner** is the owner of the horse
- **Location** is where the horse is situated,
- **Contact** is where any contact information from the owner, caretaker or where the horse is located is put.

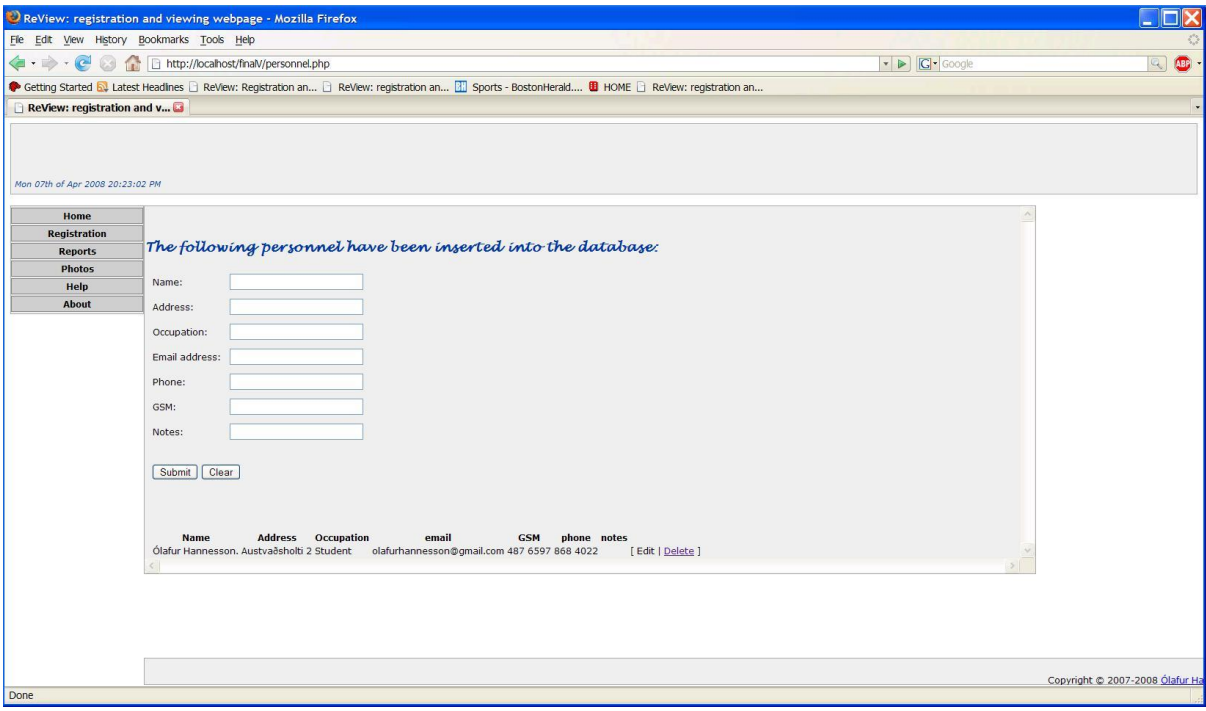
c. Colour

The *Colour* form is the simplest. The user has to write in the name of the colour and submit it. If he wants to, he can write a short description of the colour.



Picture 8: Colours

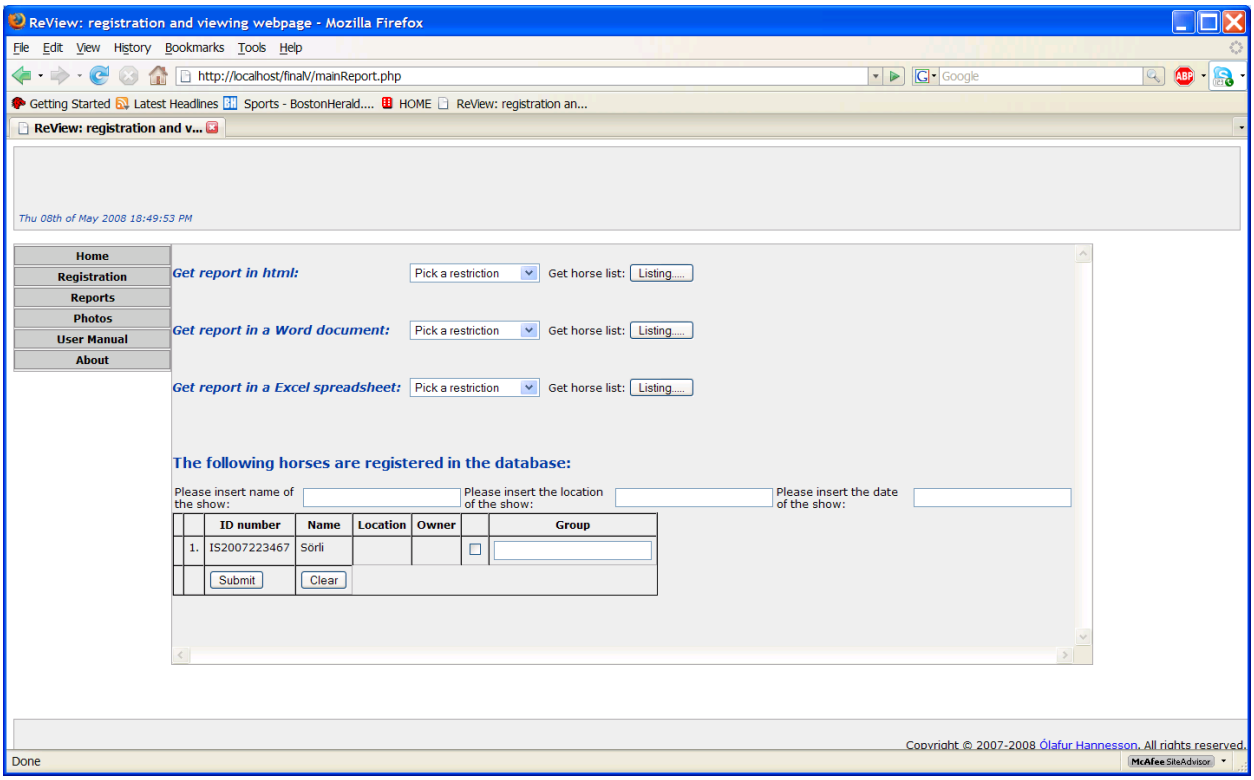
d. Personnel



Picture 9: Personnel

The *Personnel* file allows the user to register personnel, usually an owner(s) of a horse. The form is straight forward; the user fills in, the name, address, occupation, and contact information of the person. That person will then appear in a list as is shown on the page.

5. Reports

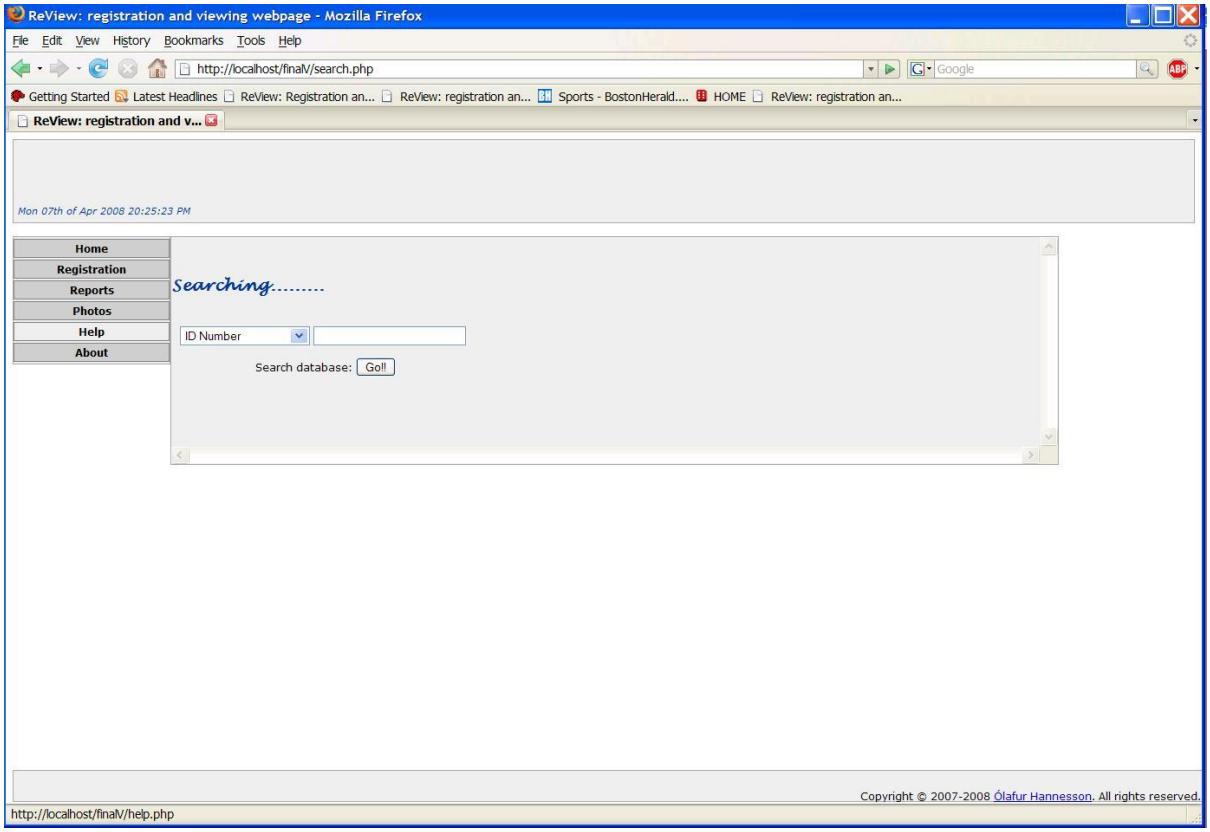


a. Reports

The user can fill in the name of the show the horses are going to be at, where the show is located and set when the date the show is to be. The user then picks the horses that are to be on the show (in the checkbox) and if the user wants, puts which group the horse is to be in. Upon submitting the form, the user can then view the report and print it out from the browser.

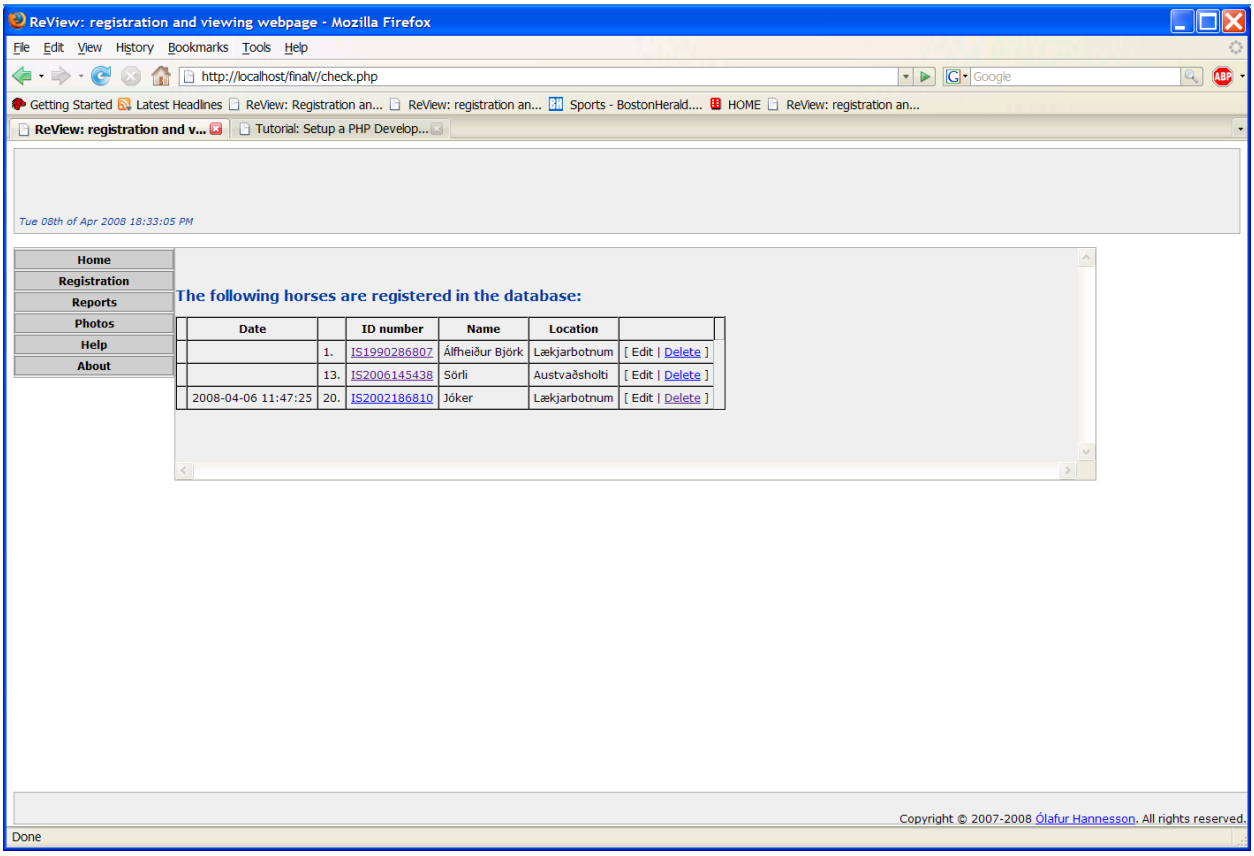
The user can pick in which order the user get the list of horses and get it in html form, word document or excel spreadsheet.

b. Search



When the user has a name of a horse, an ID number, the user can pick out which of the information he has and write in a single string. However in terms of names or location, the searching string is for single string, meaning that 'Björk' would return results while 'Björk from Haukadal' would not.

c. ´Take a look´



This page is possible the one that is most used of all. On this page the user can see which horses are registered, and when as well as view them once clicking the ID Number When the user clicks the ID number of a horse, the user is shown all the information about that particular horse.

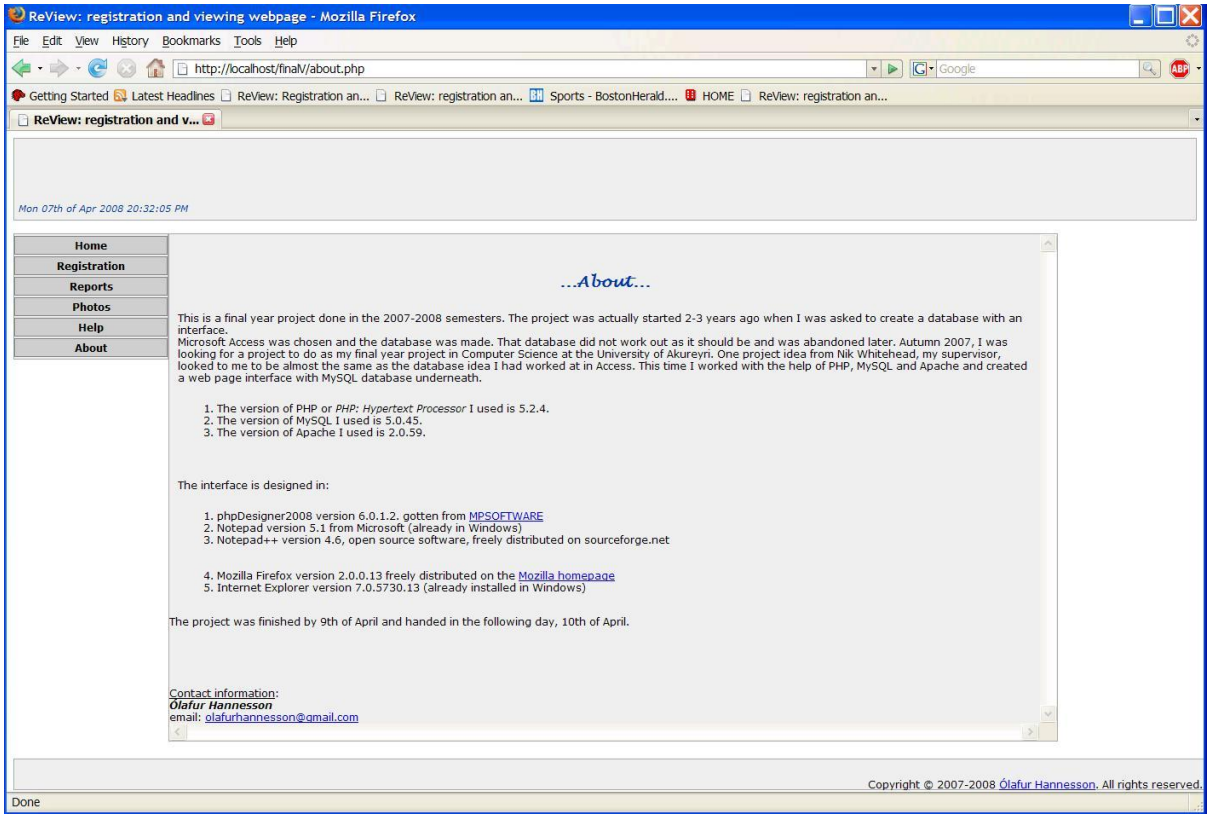
The user can delete horses from this list as well. If the user clicks on delete at the end of each line, the horse will be deleted from the list as well as the database and cannot be undone (the user must register the horse again if necessary to get it back to the list).

- **Date** is when the horse was registered in the database,
- **ID number** is the ID for the horse,
- **Name** is the name of the horse
- **Location** is where the horse is usually located
- *Delete* will delete the horse from lists and database

6. Help

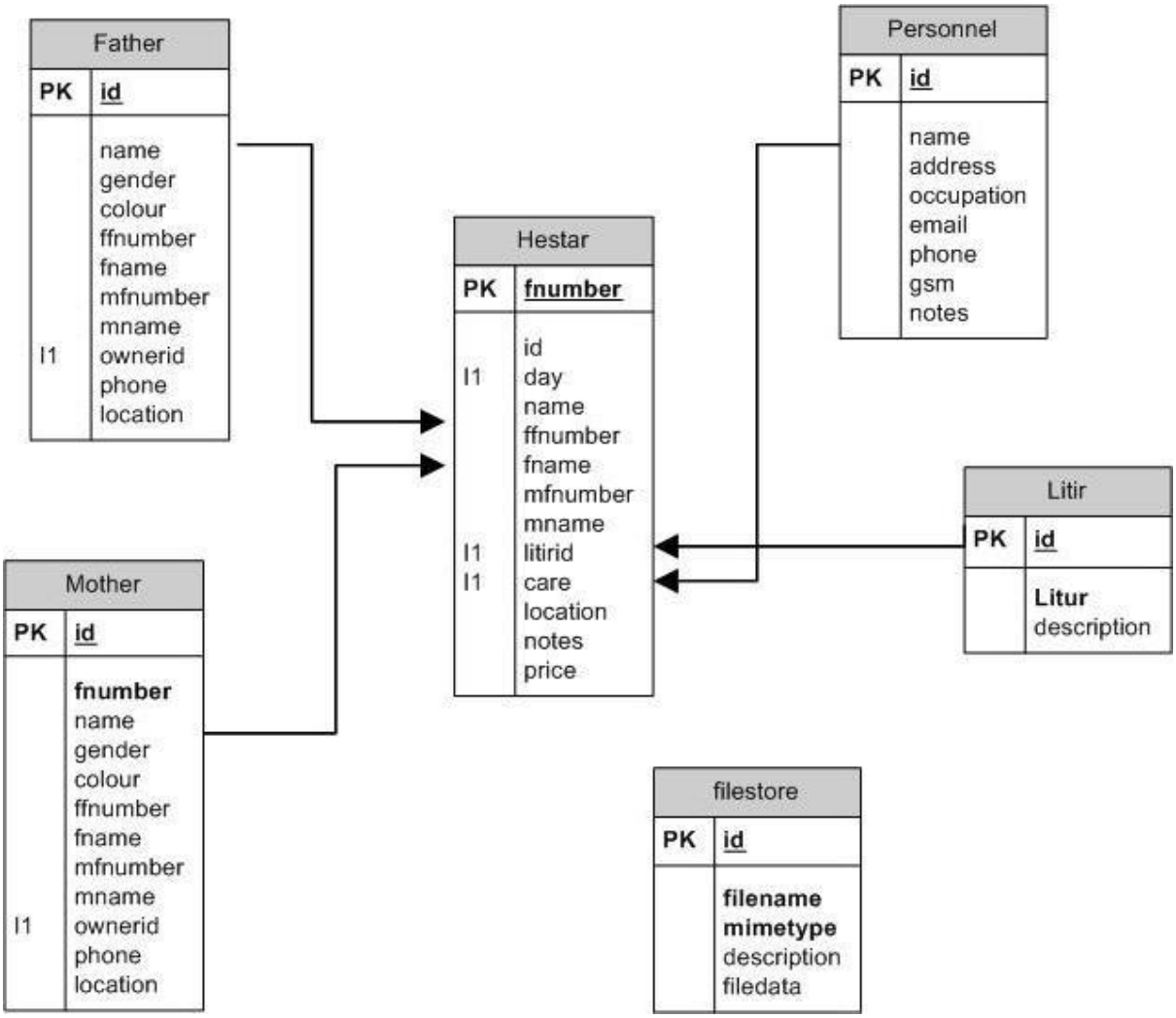
The current document is the help.

7. About



About is simply about the project and short description of the project, with lists of which programs used and programming packages.

8. The database



This is the database. ´hestar´ is the main table and gets data that has been registered into other tables and puts it into the table. ´filestore´ is table for photos.

9. References

1. Ahola, K., & Lorange, J. B. (23. July 2001). Worldfengur - ready for the world. *Information to the breeding leaders of the FEIF countries* . Europe and North America: FEIF.

2. Behlendorf, B., & al, e. (2002). *The Apache HTTP Server Project*. Sótt 8. November 2007 frá About the Apache HTTP Server Project: http://news.netcraft.com/archives/web_server_survey.html

3. Converse, T., Park, J., & Morgan, C. (2004). *PHP5 and MySQL Bible.* Indianapolis: Wiley Publishing Incorporated.

4. Finn, P., Kringeland, I., Hansen, J. S., & Antonsson, G. (2002). *FEIF Rules for Icelandic Horse Breeding.* INTERNATIONAL FEDERATION OF ICELANDIC HORSE ASSOCIATIONS.

5. NetCraft. (13. October 2007). *Web server survey.* Sótt 5. January 2008 frá Netcraft: http://news.netcraft.com/archives/web_server_survey.html

6. Pfaffenberger, B., Schafer, S., White, C., & Karow, B. (2004). *HTML, XHTML, and CSS Bible.* Indianapolis: Wiley Publishing Inc.

7. photographer, U. *Picture of Icelandic horse - Vima from Lækjabotnum.*

8. *PHP: Hypertext Preprocessor*. (Last updated 25.. November 2007). Sótt 8. November 2007 frá PHP: Preface - Manual: http://www.php.net/manual/en/preface.php

9. Ratschiller, T., & al, e. (2007). *phpMyAdmin 2.11.1 Documentation.* http://www.phpmyadmin.net/.

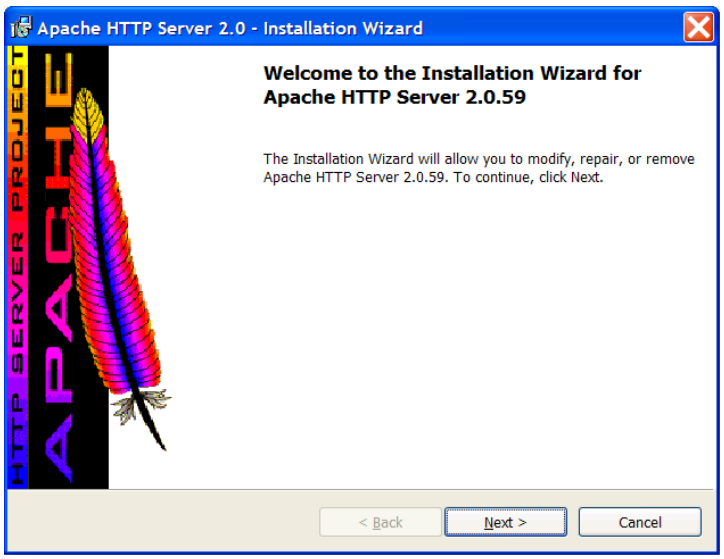
10. Yank, K. (2004). *Build your own database driven website using PHP and MySQL.* Melbourne: Sitepoint.

A. Installation of Apache

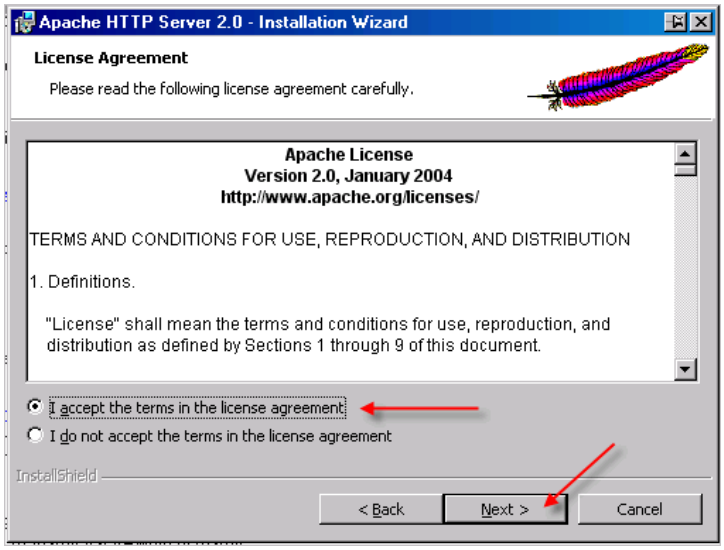
I will assume that the user has downloaded the apache 2.0.59-win32-x86-no_ssl.msi file from http://httpd.apache.org/download.cgi

1. When you have double clicked on the file, you see this window. Here you just choose Next.

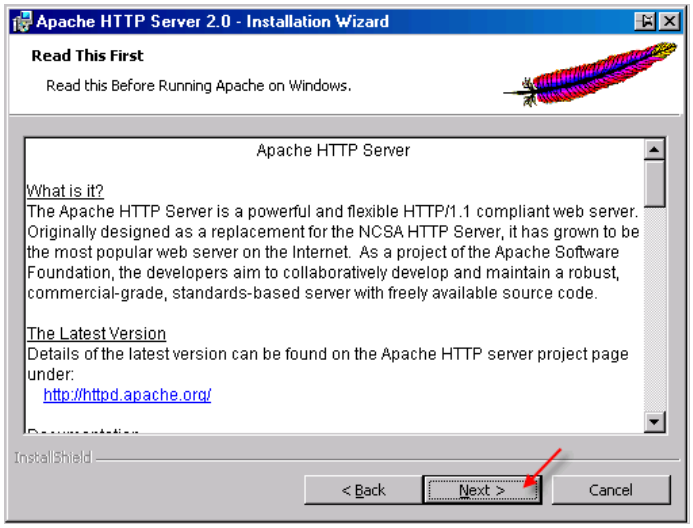
2.



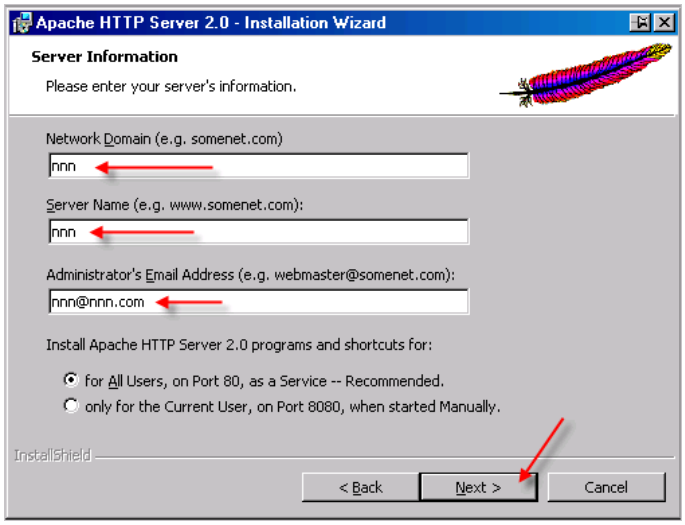
3. In this window you accept the terms in the license agreement (after having read it) and click next.


Apache HTTP Server 2.0 - Installation Wizard
License Agreement
Please read the following license agreement carefully.
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
I accept the terms in the license agreement
I do not accept the terms in the license agreement
InstallShield
< Back
Next >
Cancel

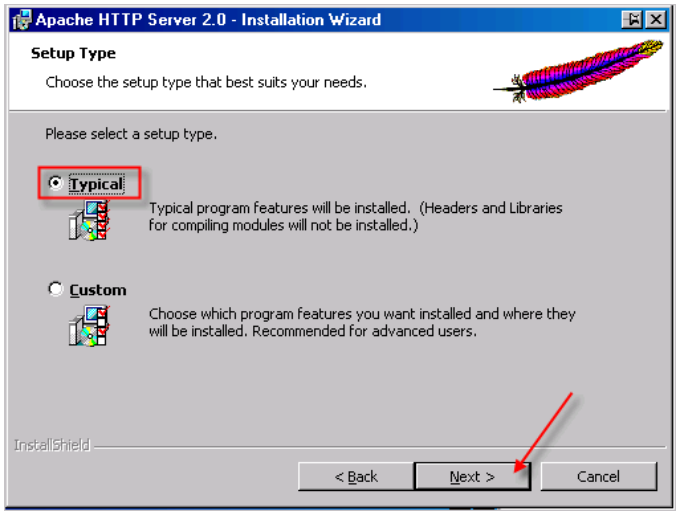
4. You read the text in this window and click next



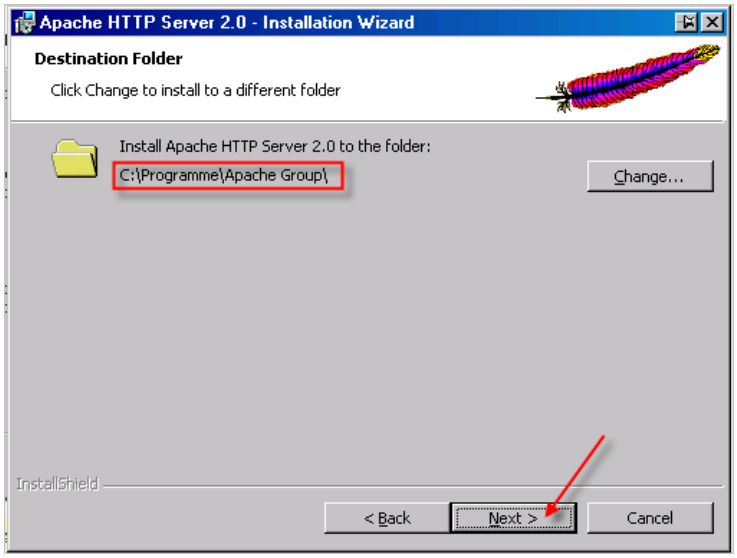
5. You fill in the boxes and click next. It does not matter what is put there but I personally put *localhost* under Network Domain and Server Name and my email address under Administrator email address. (please note that the text put in there is for demonstration purposes)



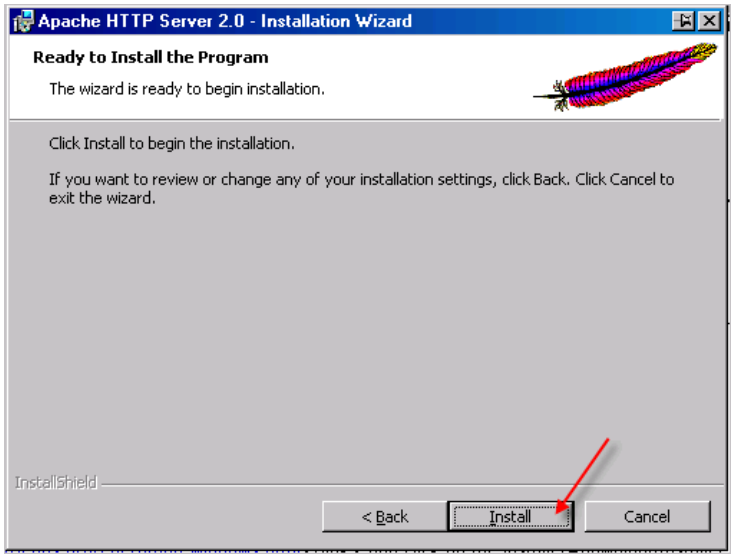
6. Here you leave *Typical* marked and click next



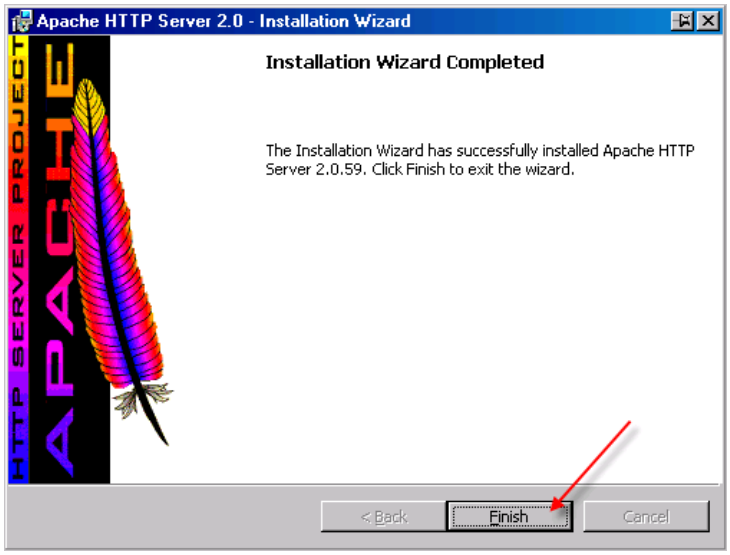
7. Here you can pick whatever location you want but best is to leave it as it is



8. Click *Install* to begin the installation



9. Click Finish



10. Should Windows XP decide to block Apache from working, a window will pop up. Click Unblock to allow Apache to interact with the firewall

11. If you go into your browser and type in http://localhost/ you should get a window stating that your installation was successful.

B. Installation of PHP

The assumption here is that the user has already downloaded PHP 5.2.4 .zip package.

1. Unzip the content of the file into a folder named c:\php
2. open the folder and find file named **php5ts.dll**. Move this file to a folder *c:\windows*
3. create a folder (in my project it was named *c:\webs*) on C drive.
4. inside the folder *c:\php* is a file named **php.ini-dist**. Rename this file to **php.ini** (right click on the file, choose rename, type php.ini)
5. Open the file and make the following changes to it,
    i. doc_root = "c:\webs" (*c:\webs* is wathever the folder you created in step 3)
    ii. change extension dir to extension_dir = "c:\php\ext"
6. If you have not installed Apache server, please do so now. Otherwise, open the **httpd.config** file for the Apache server. The file is located at *C:\Program Files\Apache Group\Apache2\conf\httpd.conf*.
    i. First of all you must find were it says DocumentRoot and change it to DocumentRoot "C:/webs", note the forward slash
    ii. Next find and change Directory (should be whatever you set DocumentRoot to be) to <Directory "C:/webs">
    iii. Thirdly find “media types” and add the following text after the 2 AddType lines,

        ScriptAlias /php/ "c:/php/"

        AddType application/x-httpd-php .php .php5

        Action application/x-httpd-php "/php/php-cgi.exe"

        SetEnv PHPRC "C:/php"

    iv. Lastly find and change the DirectoryIndex to

DirectoryIndex index.htm index.html index.php

7. After these changes please restart the Apache server by double click on the Apache icon in your taskbar and click restart
8. Now you can create your first php file and test it. You can do so by creating a file named phpinfo in the *c:\webs* folder and then in your browser call the file by typing in *http://localhost/webs/phpinfo.php*, the content of the php file should be <?php phpinfo(); ?>

C. Installation of MySQL

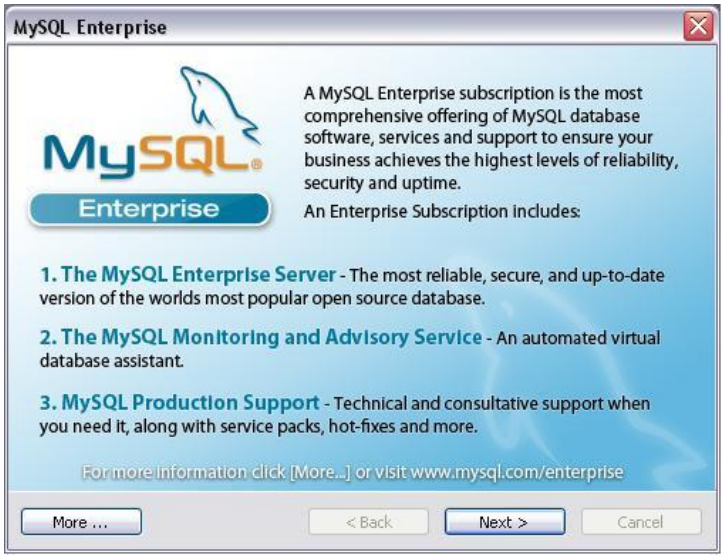
The installation of MySQL has the assumption that the user has already downloaded the mysql-essential-5.0.45-win32 file from http://dev.mysql.com/downloads/mysql/5.0.html

1. Click on Next

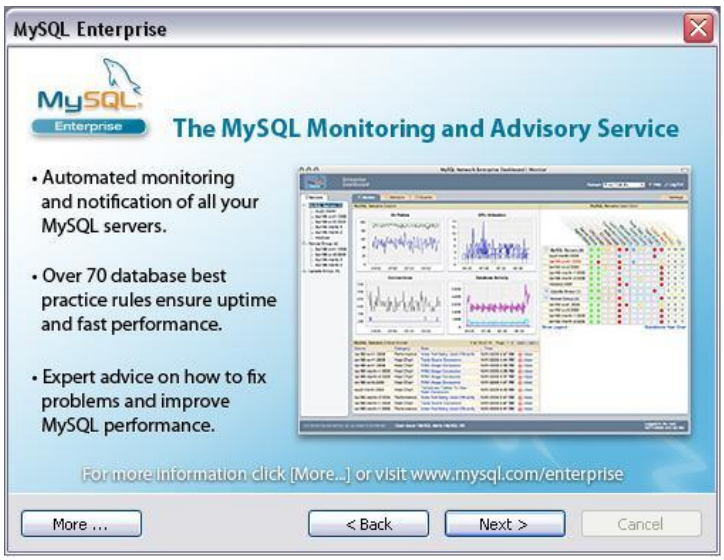
2. Select *Typical* and click next

3. Click Install

4. Click Next

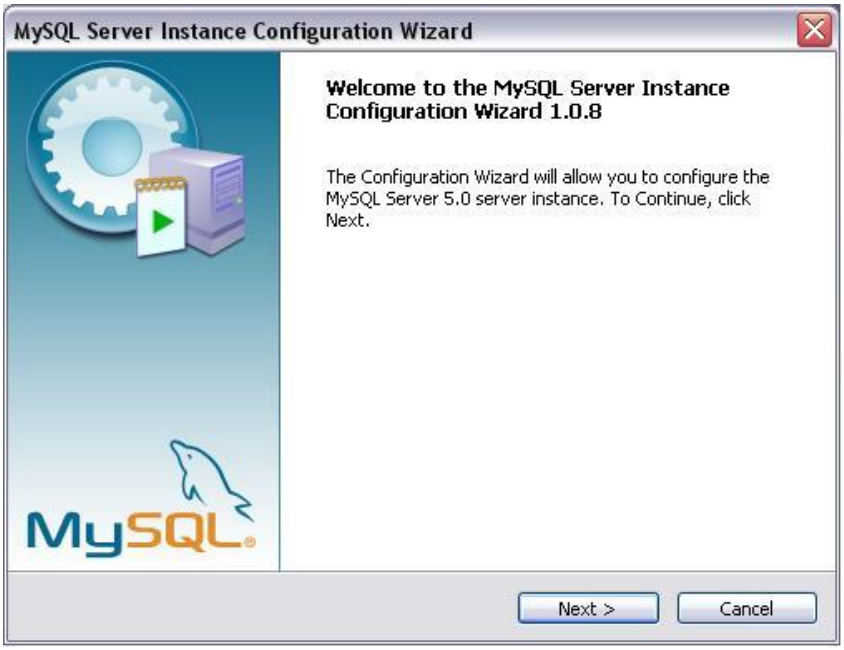


5. Click Next



6. Leave the check on for *Configure the MySQL Server now* and click next

7. Click Next

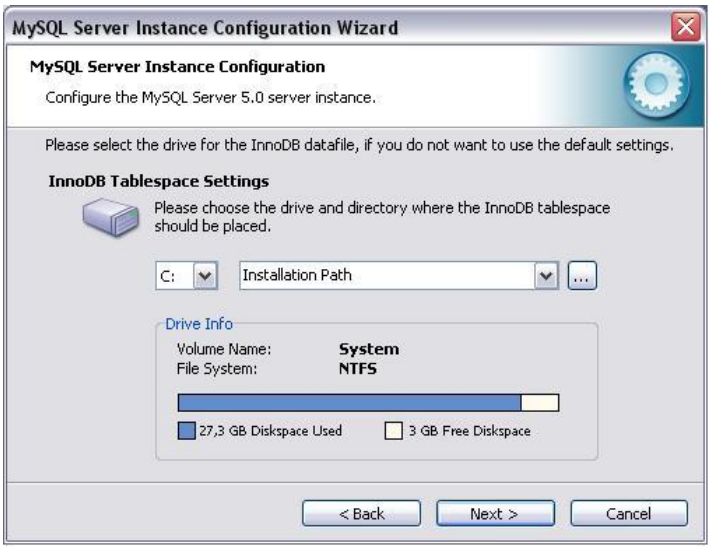


8. Select *Detailed Configuration* and click next

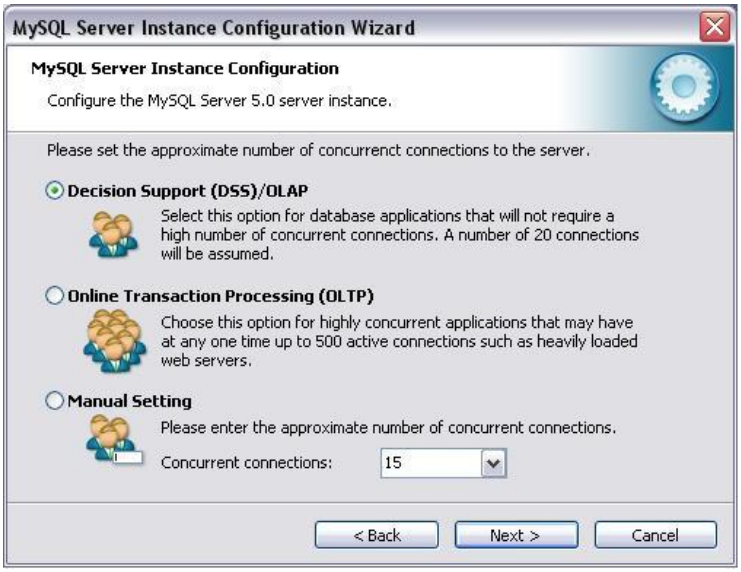
9. Select *Developer Machine* and click Next

10. Select *Multifunctional Database* and click Next

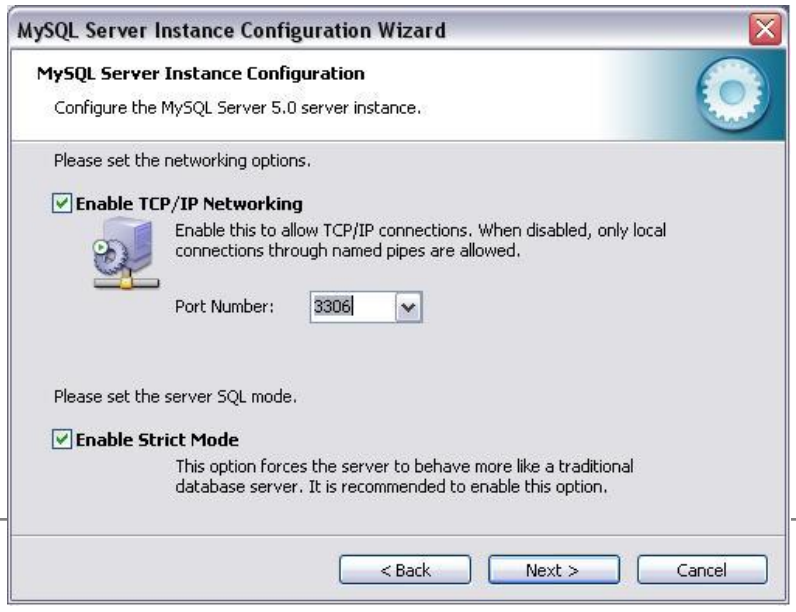
11. Select *C:* and *Installation Path* and click Next



12. Select *Decision Support (DSS)/OLAP* and click Next



13. Select *Enable TCP/IP Networking* and keep the port number on 3306 and select *Enable Strict Mode* and click Next



14. The character set is by the choice of any user but I chose to select *Best Support For Multilingualism* and click Next

15. Select *Install as Windows service,* and pick the service name from the drop down list and select *Include Bin Directory in Windows PATH* and click Next

16. Because this is for development purposes as we picked earlier in step 9, we pick some easy password (for example 'admin'). While many recommend selecting the option *Enable root access from remote machines* I did not select that option. The user needs to decide for himself on that, click Next

17. Click execute

18. Click Finish on the last window.

19. There need to be some changes made in the php.ini file. When the file has been opened, you need to find lines where are semicolons in front of the word *extension* and under there you need to find the lines *php_mysql.dll, php_mysqli.dll* and *php_pdo_mysql.dll* and remove the semicolons from the line.

20. Then you restart the Apache server by double click on the icon in the task bar and click on the restart button.

D. Installation of phpMyAdmin

The assumption of this is that the user has already downloaded a version of phpMyAdmin

1. Please unzip the file into a folder in the folder you created (c:\webs\phpMyAdmin)
2. Create a file named config.inc.php
3. Insert into the file

```
<?php
/* Servers configuration */
$i = 0;
/* Server localhost (config:root) [1] */
$i++;
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['extension'] = 'mysql';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'admin';
?>
```

4. If you go to your browser and type in http://localhost/phpMyAdmin, you should see the phpMyAdmin start screen