

# Developing a Web Application Using:

- **Dreamweaver**
- **PHP**
- **And MySQL**

STC's 53rd Annual Conference 2006

Mary Ann Howell  
Hikari Publishing © 2006  
www.hikaripub.com

## Intended Audience:

Non-programming type people who have never used PHP or MySQL, but might be enticed into it.

## Overview:

- Demo: The DocTrak web application
- Defining a web application
- The Open-Source (free) Tools I used
- Designing the database
- Designing the User Interface  
(with Dreamweaver, XHTML, PHP, and CSS)
- Advice for designing your project

## Demonstrations include:

- The DocTrak web application
- Building a MySQL database and a table with phpMyAdmin
- Adding a page in Dreamweaver that retrieves and adds information to and from the database

## Is it for you?

Maybe. If you're the type that:

- Not only enjoys building Lego projects, but also enjoys sorting and putting away the pieces.
- Likes spider solitaire so much you had to take it off your machine.
- Loves to crochet snowflakes with no. 10 yarn.
- Loved building machines with your erector set, but it's not around anymore.
- Wants to take over the world.

Or this may be information you will pass on to someone like that in your department.

## Nothing up my sleeves

- I am not selling anything.
- I'm a tech pubs supervisor; I have no connection with the software tools involved or publishing companies.

# DocTrak Demonstration

- Used in our Intranet, without security
- You can set up passwords and add security if you like

## DocTrak Functions

Has four different portals with functions specialized for each user type:

- company employees
- writers
- supervisor/scheduler
- database administrator

Functions include:

- Document searches
- List (and open) documents of a particular type for a particular model of machine
- Track the progress of documents during the review cycle
- Project scheduling, tracking, and editing
- Adding new documents
- Editing schedules
- Editing the information structure

## Story

Background I came with:

- Dreamweaver casual user
- Two beginner's PHP projects completed: one-table surveys

Our company used an Excel file for tracking documents and writer assignments

## What worked with Excel

- Our original Excel file was well designed
- The data needed to efficiently categorize documents was captured and tracked

## What didn't work

- The file was complex and looked a little scary (users looking for a document would come to me)
- It did not store history
- It did not track all the things I wanted to track
- Editing the file has to be limited to two people, to cut down on errors
- Still, I often made mistakes while editing

## So I built a web application

Some advantages of the web application:

- Users are guided as they fill in the forms
- Only data of the correct type is accepted
- If required fields are left out, the information is not added, and the user is notified
- Can track history of documents and writers' projects
- It can run on any platform

When you build your own web application, it is custom fit to your needs, instead of trying to fit your needs to an existing program.

It is easier to design a program that fits you, one customer, than it is to design a program to fit a whole bunch of unknown customers.

My ulterior motives:

- Review tracking: on whose desk is that document buried?
- Keep track of who is working on what
- Encourage edits and checks of documents
- I needed to gather time data to use later for estimating the length for time documents require and their cost
- I wanted to learn new skills

Rollout:

- The web application is modular, so I can add useful chunks as I finish them
- First edition: deployed from my website
- Then moved to my desktop
- Now it's on a network server

# The toolset I used: open source

(These definitions are paraphrased from Wikipedia <http://wikipedia.org/> .)

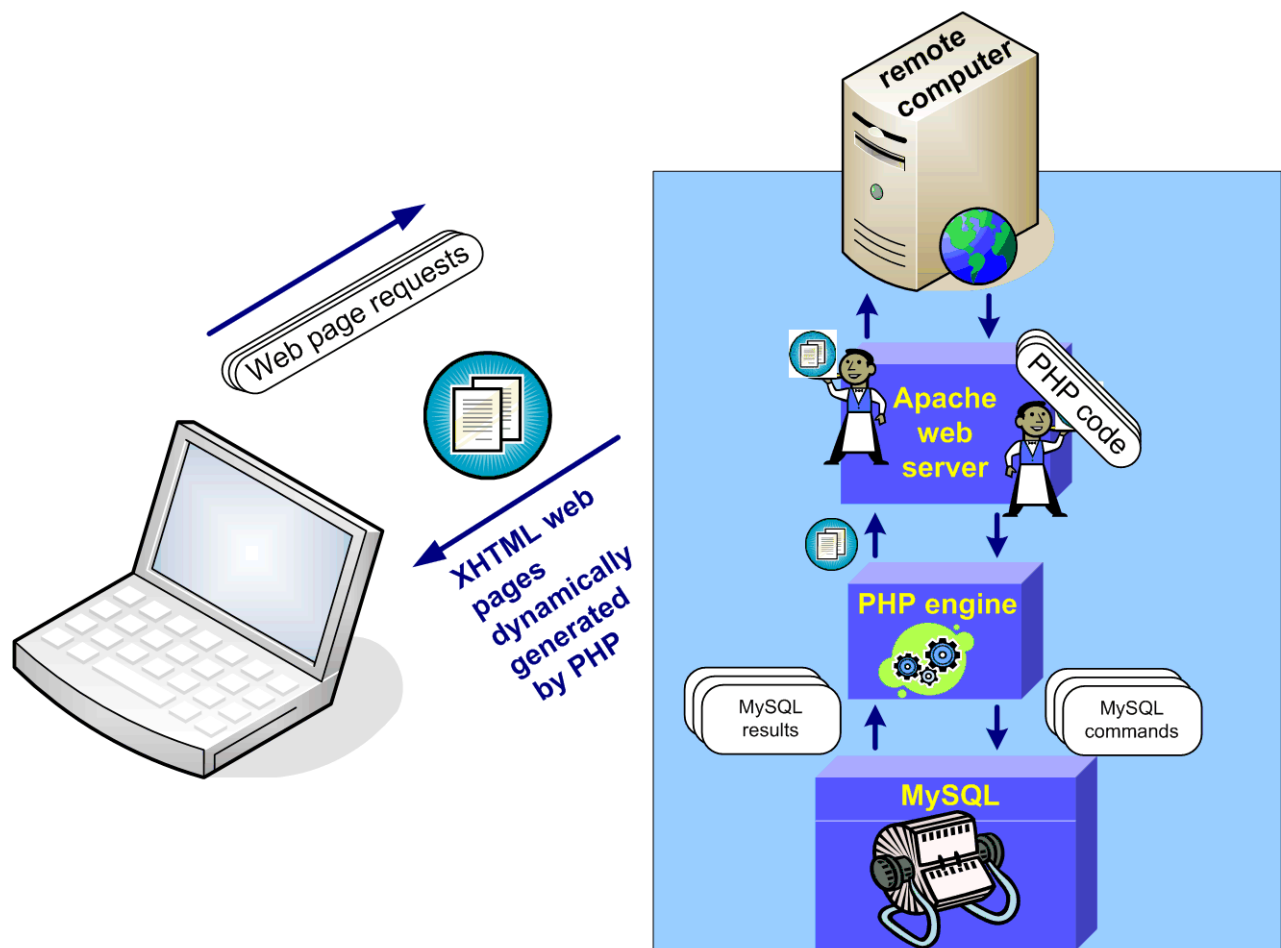
Open source tool is a term that refers to software that is created by a community rather than a vendor. The source code, often written by volunteers, is free for anyone to use or modify. (But you can't package them and re-sell them.)

Advantages:

- Software customization  
You can add a feature instead of trying to talk a vendor into doing it.
- Peer Review  
Many users and programmers review the code, reducing bugs, and keeping the software aimed at real-life usefulness.
- Longevity  
In such wide-spread use, high-quality software (which is constantly being improved) is likely to be around a long time.

## The list of tools

- Apache  
The most widely available HTTP (Hyper Text Transfer Protocol) server on the Internet. HTTP is the communications protocol that enables the Internet. It is the set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.  
Apache works with languages such as PERL and PHP.
- MySQL  
MySQL (pronounced "my ess cue el") is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database. MySQL is fast, reliable, and flexible.
- PHP (programming language)  
PHP (Hypertext Preprocessor) allows you to create dynamic web content that interacts with databases. You can also use PHP to develop web-based software applications.
- phpMyAdmin  
A widely-used interface for creating and editing MySQL databases. phpMyAdmin itself is written in PHP.



## How the tools work together

When you click the link that opens the web page with the list of released documents:

1. A request from the web page (containing PHP code) is sent from your computer to the remote computer (which hosts the website and stores the database).
2. The Apache web server says, “oh, here’s some PHP”, and passes it to the PHP engine.
3. The PHP engine processes the PHP code, and sends commands to MySQL, asking for a list of the documents that is stored in the database.
4. The MySQL server responds, pulls out the list, and hands it off to the PHP engine.
5. The PHP code runs its course, writes the HTML web page with the document list, and hands the web page off to the Apache web server.
6. Apache sends the HTML web page off to your browser and you read the document list.

## Installing the tools

- You can download these open-source tools from the Internet for free.
- Each tool comes with a configuration file you edit to set it up for your own needs.
- You don’t need to know what you are doing if you can follow directions carefully, and you use the directions found in PHP for Dreamweaver 8, by David Powers.
- You can also download an integrated suite, but these are more difficult to upgrade, and edit. You will probably also find that the tools are out-of-date.

## Plan and design

- Analyze what you want to do
- Analyze what classes of users you have
- Analyze what those users want to do
- Think about how everyone will go about their tasks
- Base it on what you are using now (if you have something)

### What I wanted this tool to do:

1. Provide a list of the documents and their info
2. List the writer's projects and their status
3. Track the documents in review
4. Search for information
5. Log and organize requests
6. Document change integration (still working on this)
  - the change request lists what product models are affected
  - then selects possible documents for revision by document type, model
  - lists the selected documents with links to add-a-rev for documents
  - notifies the writer if the document is in process

### My user types and their tasks:

- Me (database administrator)  
(tech pubs supervisor)
  - edit database structure
  - edit schedules
  - track documents
- My boss
  - edit schedules
  - track documents
- Writers
  - update their project information
  - track documents
- Company employees
  - access documentation
  - update review progress
  - request:
    - document changes
    - new documents
    - printed docs or CDs

## Define the data needed to categorize documents, and perform tasks

I analyzed our Excel file that listed and defined the tech pubs documents.

The document info:

- doc number
- document title
- what products it supports
- audience (security level)
- revision level
- revision date
- document status
  - in queue to be started
  - in process
  - in review
  - released
  - obsolete
- writer
- subject matter expert
- engineering drawing numbers associated with the document
- engineering change numbers associated with the document
- keywords
- file name
- comments

I listed the information I needed to track the writer's projects:

- Writer's name
- Project title
- Completion level (first draft, adding first edits, etc.)
- Start/finish dates
- Comments
- Hours spent
- Editors who worked on it
- If a revision, a summary list of the changes

I listed the information I wanted to track the document review process:

- Document title
- Reviewer's name
- The date the reviewer got the document
- Whether the receiver signed off the document
- Comments from the reviewer
- The date the reviewer signed off the document (if signed off)

# Designing the database

## Structure

Excel file structure:

- Excel file is flat as a piece of paper.
- It is a single file of plain text.
- It is a single table that has no relationship with other tables.
- It is a spreadsheet tool designed for calculations with numbers, often very complex, and it does this very well.

Relational database structure:

- You use your browser as an interface (phpMyAdmin)
- One or more tables that have:
  - columns: fields of related data
  - rows: each row is a record
- Types of data:
  - unique key: a field in each record that contains data that is unique to that record. The record can use that unique key as its name, so it can come when called.
  - numbers
  - dates
  - text
  - binary objects

Table relationships:

- One to one: this chunk of information is used by only one other chunk of information
- One to many: this same chunk of information is used by many other chunks of information
- Many to many: multiple chunks of this information are used in multiples by multiple other chunks of information

Design strategy:

- Divide the information into separate tables, grouped by how the information functions
- Normalization: no redundant info: don't have the same information in different tables
- Low-hanging-fruit tables (easy to design): the ones that populate the drop-down lists

## MySQL commands (query language)

- Fairly simple
- Build and restructure tables
- Insert, update, and delete records
- Select data to pass on



## User interface

Do this last, and **don't get bogged down in formatting**

- The beauty of a web app, is that your users only need their browser to use your program—there is nothing to install.
- The interface is built with HTML (XHTML) web pages juiced up with PHP.
- I use Dreamweaver, which has built-in support for PHP and MySQL.

My pages:

- Are linked to a Dreamweaver template
- Use cascading style sheets
- Have a database connection
- Are stored in folders organized somewhat like my application

## PHP basics

PHP is a great way to get your feet wet as a programmer.

You'll learn to use:

- Variables (writer = Bob)
- Arrays (variables that contain a list of variables)
- Control structures (if this equals 0, stop)
- Functions
  - Functions are reusable blocks of code that are combined together to do a task.
  - You give the function a name; once it has a name, it comes when you call it.

PHP employs MySQL commands and HTML.

(It was the HTML that built all the nifty fields in the form, not PHP.)

## Where to go from here

You can practice by working on your website. Your website can feature:

- Hobbies
- Family
- Portfolio
- Rantings about whatever moves you

## Learn the tools

- If you aren't familiar with Dreamweaver and XHTML/HTML, start with a basic Dreamweaver tutorial-style book
- If you can get around in Dreamweaver, get *Foundation PHP for Dreamweaver 8* by David Powers (a Friends of Ed book)

## David Powers' Book

Average rating on Amazon: five stars

Foundation PHP for Dreamweaver 8 (Paperback) by David Powers

Paperback: 530 pages

Publisher: friends of ED (December 19, 2005)

ISBN: 159059569-6

## Why I like David's book

- I have five books with instructions on installing the open-source tools needed. This is the first time anyone's installation instructions worked for me.
- He writes in plain English with explanations that are simple and concise, but complete.
- He gives step-by-step instructions that cover each detail, and the instructions are correct.
- He provides the code and examples he uses in the book, and shows you how to install a free tool to compare your code with his, to see where you went wrong.
- He has an up-to-date errata web page posted.
- He gives almost instant feedback to you on the friends of Ed forum, if you get stuck.

## Start your project

If not at work, do it at home and during lunch time.

Pick out a domain name and get a website.

Mine is hosted by <http://www.amhosting.com/>

- Register your domain name for \$25 for a year
- Set up a website for \$8.00 per month
- Includes PHP support and 5 MySQL databases
- 200MB Storage Space

Work strategy:

- Set up a sandbox on your computer
- Install the open-source tools
- Take your time planning the database
- Plan your database carefully

## Project strategy:

- Modular: produce a chunk at a time
- These chunks are usable, so you provide value as you go
- My flow was my priority list (most desperately needed first)
  - List the documents
  - Track writer's projects
  - Provide access to documents
  - Rein in the document review cycle
- Use good project planning
  - Know your users and what they need to do
  - Plan and design the database carefully

## In conclusion

1. Stretch: it's good for you, feels good, keeps life interesting, and keeps you young.  
Learn new stuff; it's good for your brain, and keeps it exercised.
2. Open-source tools can be used to tailor fit a solution for you, at a fraction of the cost of the Big Behemoths.
3. STC: what you put in, you get back... and way more.

Mary Ann Howell

Hikari Publishing

[www.hikaripub.com](http://www.hikaripub.com)

Recommended book: Foundation PHP for Dreamweaver 8 (Paperback) by David Powers

