Lab 2.5

Flat-file database development with CGI

Objectives

After completing this lab, you will be able to:

- Implement a Web database for small office and home office (SOHO).

Synopsis

One of the major advantages of the Web is its global reach. In order to capture records of potential clients or members, small businesses and organizations or event organizers often need to implement some kind of database to facilitate the management of memberships or customer records. In this lab, you will be guided to implement a basic Web database for such purposes. The database is implemented by a plain text file. It can be accessed via the provided CGI scripts written in Perl. This lab demonstrates that we can implement a small and yet flexible database using only a flat file and a few relatively simple CGI scripts, without needing to resort to a fully fledged relational database. Such an approach is surprisingly effective, and is sufficient under many circumstances.

Prerequisites

- You have a user account with remote login privileges on the LabBook Support Server.
- You are familiar with the use of an SSH client such as PuTTY.
- Your PC is connected to the Internet.
- You are familiar with basic UNIX commands and HTML.

Background reading and preparation

Read the following online resources before doing this lab:

- ‘Perl’ — http://en.wikipedia.org/wiki/Perl

  You are encouraged to read through the whole article. However, for the purposes of this lab, you may choose to spend no more than 30 minutes skimming through it.

Expected duration

Approximately one hour (including background reading and preparation)

Procedure

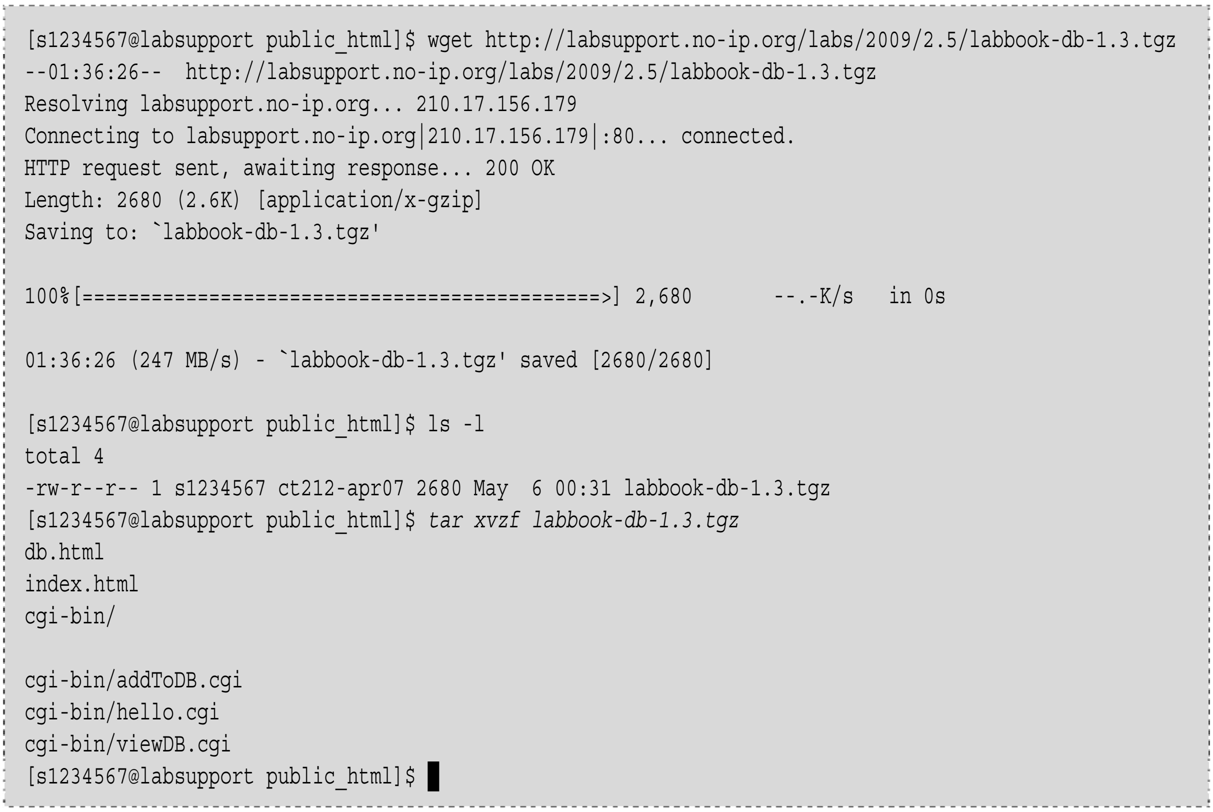
Step 1 Downloading and extracting the Web database source files

In Lab 2.2 Step 1, you verified the existence of the `public_html` directory under your home directory. There should also be a `cgi-bin` directory under `public_html`. If you have removed these two directories, create them again and set up the proper access permissions for them by following the instructions in Lab 1.8 Step 1.

The database and the associated source files can be downloaded from the LabBook Support Server. These files are compressed as a tarball (see Lab 1.6 Step 1 for information about tarballs), which is located here:

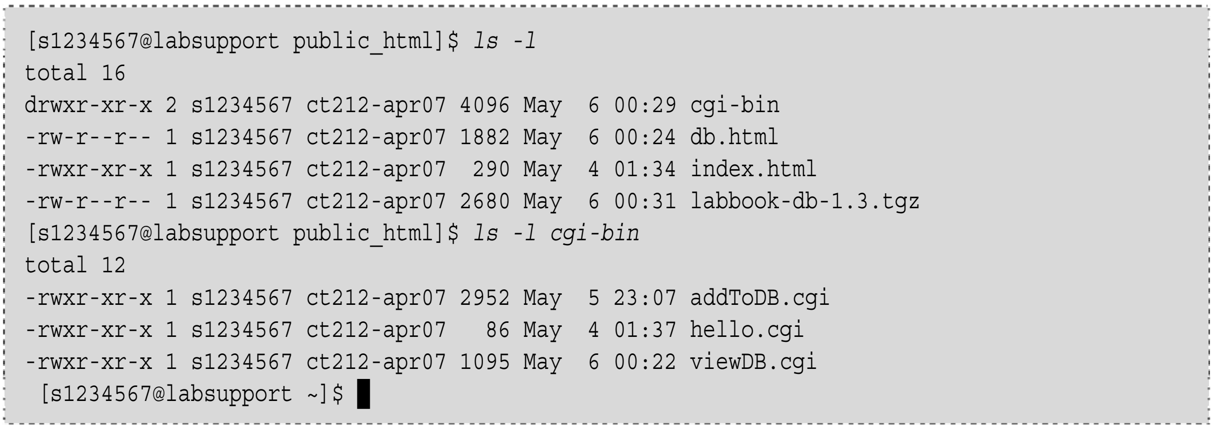
http://labsupport.no-ip.org/labs/2009/2.5/labbook-db-1.3.tgz

Change to `public_html` and use the `wget` command to download the tarball. Then use the `tar` command to extract the source files form the tarball.



```
[s1234567@labsupport public_html]$ wget http://labsupport.no-ip.org/labs/2009/2.5/labbook-db-1.3.tgz
--01:36:26--  http://labsupport.no-ip.org/labs/2009/2.5/labbook-db-1.3.tgz
Resolving labsupport.no-ip.org... 210.17.156.179
Connecting to labsupport.no-ip.org|210.17.156.179|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2680 (2.6K) [application/x-gzip]
Saving to: `labbook-db-1.3.tgz'

100%[==========================================>] 2,680       --.-K/s   in 0s

01:36:26 (247 MB/s) - `labbook-db-1.3.tgz' saved [2680/2680]

[s1234567@labsupport public_html]$ ls -l
total 4
-rw-r--r-- 1 s1234567 ct212-apr07 2680 May  6 00:31 labbook-db-1.3.tgz
[s1234567@labsupport public_html]$ tar xvzf labbook-db-1.3.tgz
db.html
index.html
cgi-bin/

cgi-bin/addToDB.cgi
cgi-bin/hello.cgi
cgi-bin/viewDB.cgi
[s1234567@labsupport public_html]$ █
```

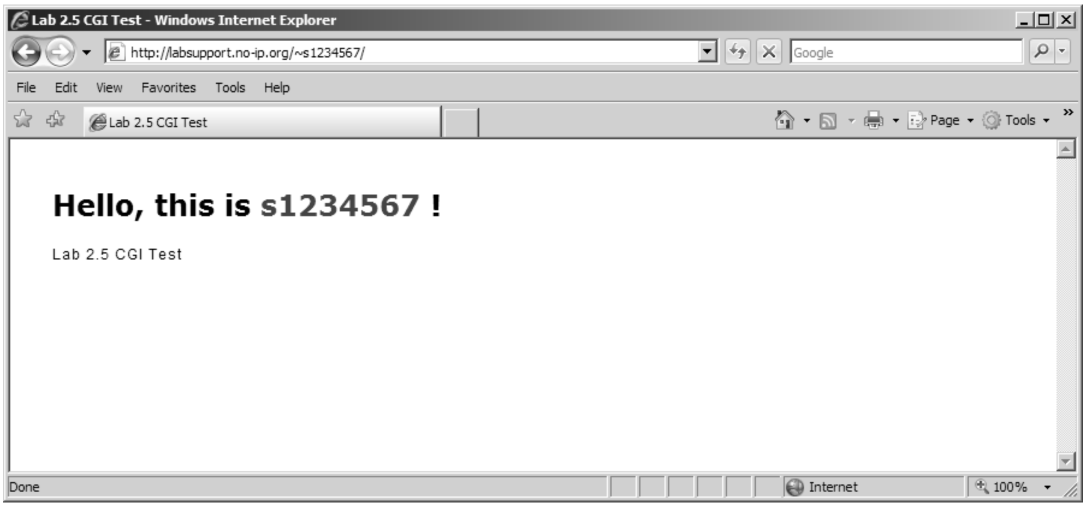
After the extraction, you should see two new files, `index.html` and `db.html`, and one new directory, `cgi-bin`, created under your `public_html` directory. Verify that `index.html` and `db.html` are readable by all users. In addition, verify that `index.html` has the execute permission enabled. (This is necessary because `index.html` utilizes the server-side include feature.) For the `cgi-bin` directory, verify that it is readable and accessible by all users. The `cgi-bin` directory contains three CGI scripts, `addToDB.cgi`, `hello.cgi` and `viewDB.cgi`. Verify that they are all readable and executable by all users. Remember that you can use the `chmod` command to change the file permissions as necessary.



```
[s1234567@labsupport public_html]$ ls -l
total 16
drwxr-xr-x 2 s1234567 ct212-apr07 4096 May  6 00:29 cgi-bin
-rw-r--r-- 1 s1234567 ct212-apr07 1882 May  6 00:24 db.html
-rwxr-xr-x 1 s1234567 ct212-apr07  290 May  4 01:34 index.html
-rw-r--r-- 1 s1234567 ct212-apr07 2680 May  6 00:31 labbook-db-1.3.tgz
[s1234567@labsupport public_html]$ ls -l cgi-bin
total 12
-rwxr-xr-x 1 s1234567 ct212-apr07 2952 May  5 23:07 addToDB.cgi
-rwxr-xr-x 1 s1234567 ct212-apr07   86 May  4 01:37 hello.cgi
-rwxr-xr-x 1 s1234567 ct212-apr07 1095 May  6 00:22 viewDB.cgi
 [s1234567@labsupport ~]$ █
```

Step 2 Checking your personal website

Visit your personal website at http://labsupport.no-ip.org/~s1234567. You should see the following output.



Lab 2.5 CGI Test - Windows Internet Explorer
http://labsupport.no-ip.org/~s1234567/
Google
File Edit View Favorites Tools Help
Lab 2.5 CGI Test
Page
Tools
Done
Internet
100%

Hello, this is s1234567 !

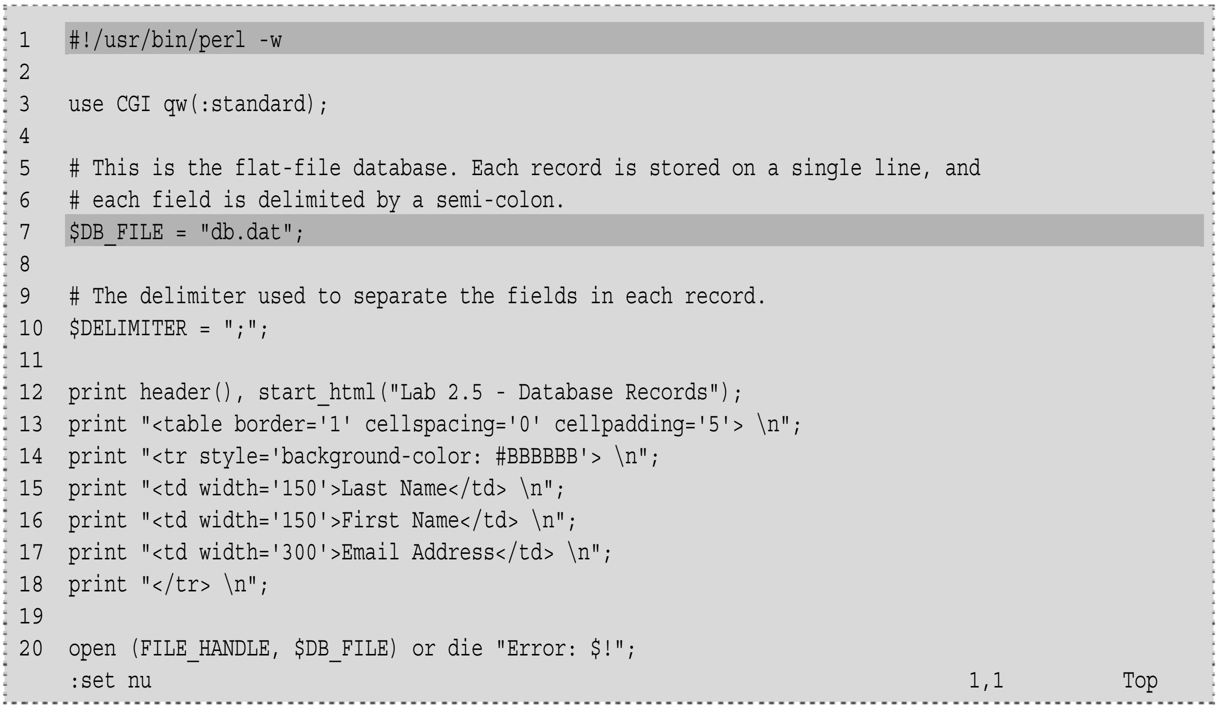
Lab 2.5 CGI Test

Figure 2.5.1

The above webpage is displayed because you have created a default webpage, `index.html`, in your homepage directory. If your username is displayed correctly in the above webpage, then CGI is working properly on your personal website.

Step 3 Configuring the Web database program

Our Web database program consists of two CGI scripts, `viewDB.cgi` and `addToDB.cgi`, which are both written in Perl. `viewDB.cgi` is used for reading the records in the database while `addToDB.cgi` is for adding new records into it. The database is implemented as an ASCII plain-text file. Change to the directory `cgi-bin` and open the file `viewDB.cgi` using vi or your favourite text editor.



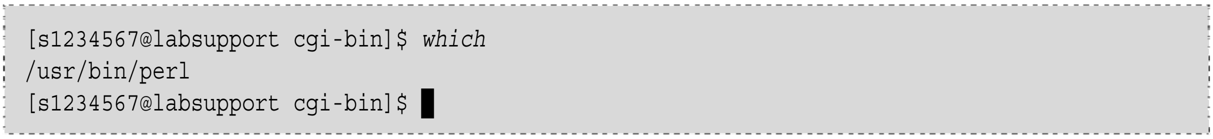
```
1   #!/usr/bin/perl -w
2
3   use CGI qw(:standard);
4
5   # This is the flat-file database. Each record is stored on a single line, and
6   # each field is delimited by a semi-colon.
7   $DB_FILE = "db.dat";
8
9   # The delimiter used to separate the fields in each record.
10  $DELIMITER = ";";
11
12  print header(), start_html("Lab 2.5 - Database Records");
13  print "<table border='1' cellspacing='0' cellpadding='5'> \n";
14  print "<tr style='background-color: #BBBBBB'> \n";
15  print "<td width='150'>Last Name</td> \n";
16  print "<td width='150'>First Name</td> \n";
17  print "<td width='300'>Email Address</td> \n";
18  print "</tr> \n";
19
20  open (FILE_HANDLE, $DB_FILE) or die "Error: $!";
    :set nu                                                          1,1         Top
```

Listing 2.5.1

Notice line 1 of the script:

`#!/usr/bin/perl -w`

This line indicates that the Perl script is intended to be executed by the Perl interpreter specified by the full path `/usr/bin/perl`. On most Linux systems, the Perl interpreter is installed in the `/usr/bin` directory, and so it is not necessary to change this line. You can use the `which` command to check the location of the Perl interpreter on the LabBook Support Server.



```
[s1234567@labsupport cgi-bin]$ which
/usr/bin/perl
[s1234567@labsupport cgi-bin]$
```

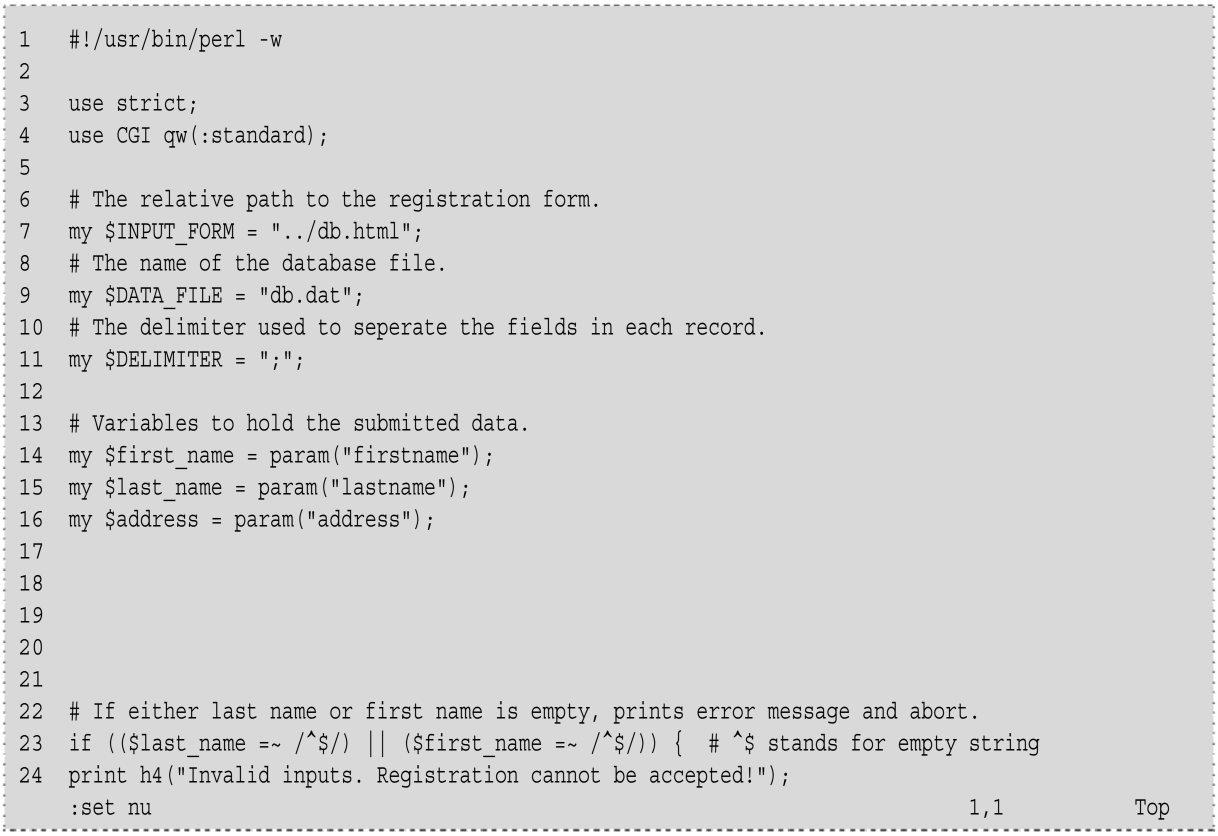
If you find the location of the Perl interpreter to be different, you will need to modify the first line of both `viewDB.cgi` and `addToDB.cgi` accordingly.

Next, notice line 7 of the script:

`$DB_FILE = "db.dat";`

The variable `$DB_FILE` holds the name of the database file that we are going to use, which is `db.dat`. There is no need to change this file name, but in case you want to, you need to change the value of `$DB_FILE` in both `viewDB.cgi` and `addToDB.cgi`. You are encouraged to read the rest of the script and try to understand what the script is intended to do.

Next, open and study the script addToDB.cgi.



```
#!/usr/bin/perl -w

use strict;
use CGI qw(:standard);

# The relative path to the registration form.
my $INPUT_FORM = "../db.html";
# The name of the database file.
my $DATA_FILE = "db.dat";
# The delimiter used to seperate the fields in each record.
my $DELIMITER = ";";

# Variables to hold the submitted data.
my $first_name = param("firstname");
my $last_name = param("lastname");
my $address = param("address");





# If either last name or first name is empty, prints error message and abort.
if (($last_name =~ /^$/) || ($first_name =~ /^$/)) {  # ^$ stands for empty string
print h4("Invalid inputs. Registration cannot be accepted!");
:set nu                                                                 1,1          Top
```

Listing 2.5.2

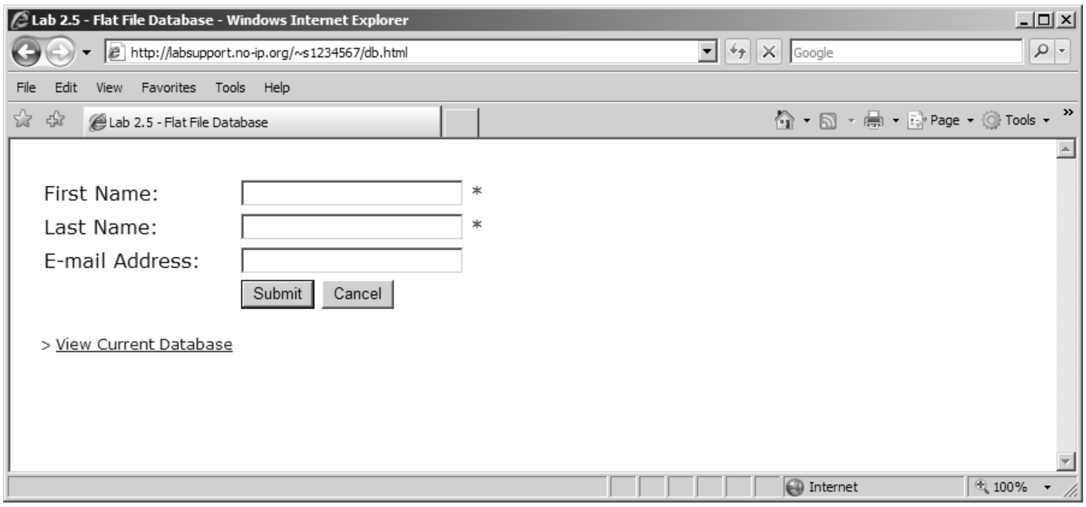
Again, you don't need to change anything in this script.

Step 4 Entering data into the Web database

You are now ready to test your Web database. Visit this URL:

http://labsupport.no-ip.org/~s1234567/db.html

You will see the following registration form.

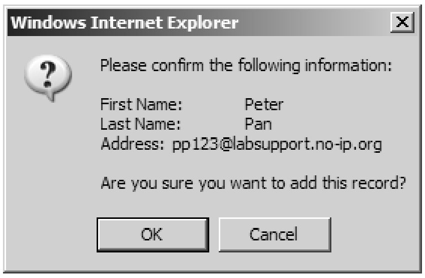

Lab 2.5 - Flat File Database - Windows Internet Explorer
http://labsupport.no-ip.org/~s1234567/db.html
Google
File
Edit
View
Favorites
Tools
Help
Lab 2.5 - Flat File Database
Page
Tools
Internet
100%

First Name: *
Last Name: *
E-mail Address:
Submit Cancel

> View Current Database

Figure 2.5.2

Enter some test data in the form and press the **Submit** button. The JavaScript embedded on the registration form will ask you to confirm your input. Press the **OK** button to confirm.



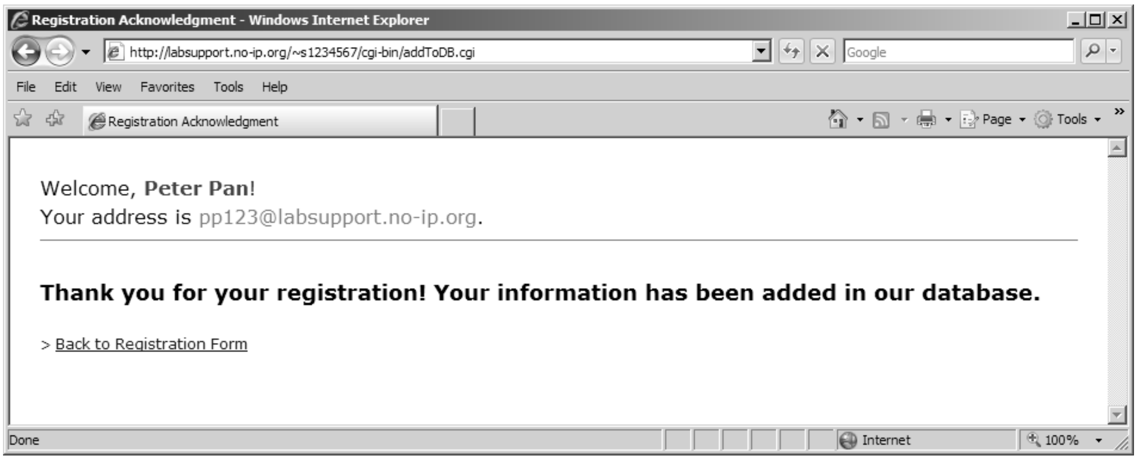
Please confirm the following information:

First Name: Peter
Last Name: Pan
Address: pp123@labsupport.no-ip.org

Are you sure you want to add this record?

Figure 2.5.3

Your registration should be accepted without any problem.


Registration Acknowledgment - Windows Internet Explorer
http://labsupport.no-ip.org/~s1234567/cgi-bin/addToDB.cgi
Google
File
Edit
View
Favorites
Tools
Help
Registration Acknowledgment
Page
Tools
Done
Internet
100%

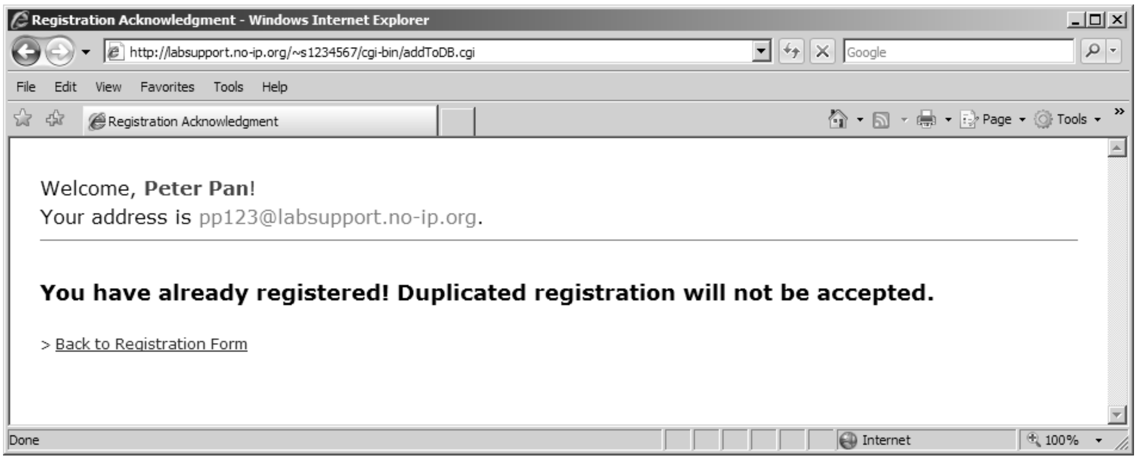
Welcome, **Peter Pan**!
Your address is pp123@labsupport.no-ip.org.

Thank you for your registration! Your information has been added in our database.

> Back to Registration Form

Figure 2.5.4

Our Web database does not allow the same person to register more than once. Try entering the same data on the registration form and submit again. This time the registration will be rejected.


Registration Acknowledgment - Windows Internet Explorer
http://labsupport.no-ip.org/~s1234567/cgi-bin/addToDB.cgi
Google
File Edit View Favorites Tools Help
Registration Acknowledgment
Page
Tools
Done
Internet
100%

Welcome, **Peter Pan**!
Your address is pp123@labsupport.no-ip.org.

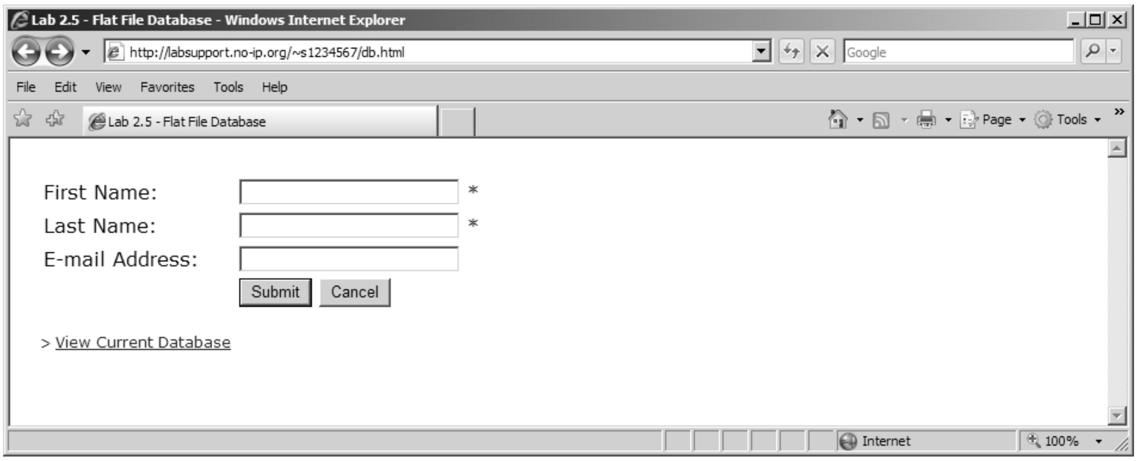
You have already registered! Duplicated registration will not be accepted.

> Back to Registration Form

Figure 2.5.5

Step 5 Viewing the records in your Web database

On the registration form there is a link at the bottom of the page: **View Current Database**.

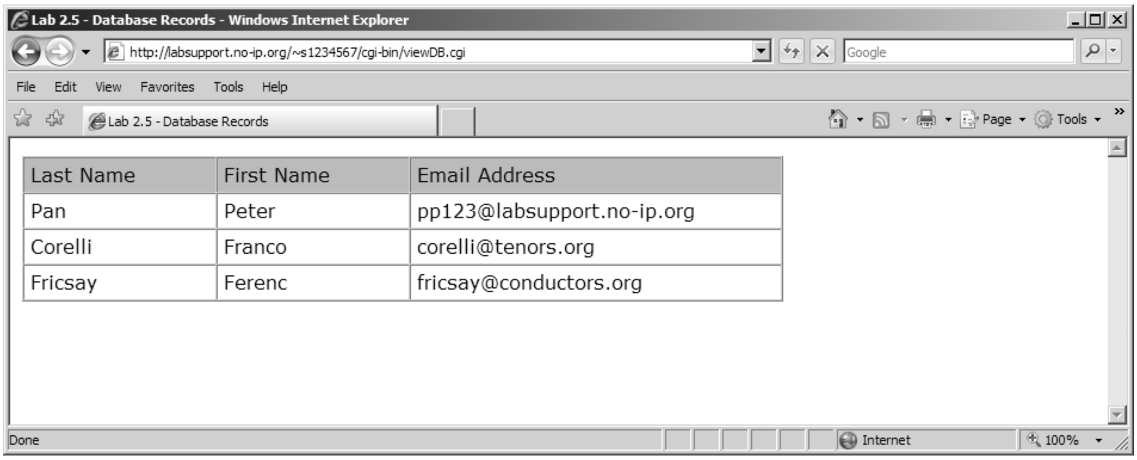

Lab 2.5 - Flat File Database - Windows Internet Explorer
http://labsupport.no-ip.org/~s1234567/db.html
Google
File Edit View Favorites Tools Help
Lab 2.5 - Flat File Database
Page
Tools
Submit
Cancel
Internet
100%

First Name: *
Last Name: *
E-mail Address:

> View Current Database

Figure 2.5.6

Click the link and the records in the database will be displayed.



Lab 2.5 - Database Records - Windows Internet Explorer
http://labsupport.no-ip.org/~s1234567/cgi-bin/viewDB.cgi
Google
File Edit View Favorites Tools Help
Lab 2.5 - Database Records
Page Tools
Done
Internet
100%

Last Name	First Name	Email Address
Pan	Peter	pp123@labsupport.no-ip.org
Corelli	Franco	corelli@tenors.org
Fricsay	Ferenc	fricsay@conductors.org

Figure 2.5.7

You have successfully finished this lab.

Summary

In this lab, you learned how to implement a small and yet flexible database for use on a website. A database is the soul of a dynamic website, and Web databases are very useful even for small or home office use. You also became acquainted with CGI scripts written in Perl.

Questions and exercises

1 Can you think of two additional functionalities that can enhance the Web database you have just implemented?

2 Set up password protection so that only authorized users with correct passwords can view the records.

3 So far, new entries are appended one by one with the latest entry appearing at the end of the list. Can this be changed easily so that the latest entry always appears first in the list?

Mini-projects

Note: Be prepared to spend up to eight hours (perhaps even more) doing these mini-projects. Good luck!

1 Modify the Perl-CGI script `viewDB.cgi` so that records can be viewed in either alphabetical or reverse order.

2 Modify the input form `db.html` to become a formal registration form that one often sees on a typical conference registration webpage. Incorporate the following functionalities: password protection; and add, delete and sort records in your revised registration database.