CHAPTER 4

Relational Database Management System: MySQL

This chapter introduces the student to the MySQL database management system and PHP, the programming language used to program applications that access a MySQL database. The discussion in this chapter is not specific to any version of MySQL and all examples would work with MySQL 4.0 or higher version.

The `COMPANY` database of the Elmasri/Navathe text is used throughout this chapter. In Section 4.1, a larger data set is introduced for the `COMPANY` database. In Section 4.2, the `mysql` utility is introduced which allows the users to interact with the database including running commands, executing SQL statements, and running MySQL scripts. In Section 4.3 PHP programming with MySQL is introduced through a complete Web browsing application for the `COMPANY` database. Finally, in Section 4.4 an online address book application is discussed with interfaces for adding, deleting, listing and searching a collection of contacts coded in HTML as well as in PhP.

4.1 COMPANY Database

Consider the COMPANY database state shown in Figure 5.6. Let us assume that the company is expanding with 3 new departments: Software, Hardware, and Sales. The Software department has 2 locations: Atlanta and Sacramento, the Hardware department is located in Milwaukee, and the Sales department has 5 locations: Chicago, Dallas, Philadelphia, Seattle, and Miami.

The `Software` department has 3 new projects: `OperatingSystems`, `DatabaseSystems`, and `Middleware` and the `Hardware` department has 2 new projects: `InkjetPrinters` and `LaserPrinters`. The company has added 32 new employees in this expansion process.

The updated `COMPANY` database is shown in the following tables.

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	22-May-78
	Administration	4	987654321	1-Jan-85
	Headquarters	1	888665555	19-Jun-71
	Software	6	111111100	15-May-99
	Hardware	7	444444400	15-May-98
	Sales	8	555555500	1-Jan-97

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4
	OperatingSystems	61	Jacksonville	6
	DatabaseSystems	62	Birmingham	6
	Middleware	63	Jackson	6
	InkjetPrinters	91	Phoenix	7
	LaserPrinters	92	LasVegas	7

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston
	6	Atlanta
	6	Sacramento
	7	Milwaukee
	8	Chicago
	8	Dallas
	8	Philadephia
	8	Seattle

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATION
	333445555	Alice	F	5-Apr-76	Daughter
	333445555	Theodore	M	25-Oct-73	Son
	333445555	Joy	F	3-May-48	Spouse
	987654321	Abner	M	29-Feb-32	Spouse
	123456789	Michael	M	1-Jan-78	Son
	123456789	Alice	F	31-Dec-78	Daughter
	123456789	Elizabeth	F	5-May-57	Spouse
	444444400	Johnny	M	4-Apr-97	Son
	444444400	Tommy	M	7-Jun-99	Son
	444444401	Chris	M	19-Apr-69	Spouse
	444444402	Sam	M	14-Feb-64	Spouse

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40
	453453453	1	20
	453453453	2	20
	333445555	2	10
	333445555	3	10
	333445555	10	10
	333445555	20	10
	999887777	30	30
	999887777	10	10
	987987987	10	35
	987987987	30	5
	987654321	30	20
	987654321	20	15
	888665555	20	null
	111111100	61	40
	111111101	61	40
	111111102	61	40
	111111103	61	40
	222222200	62	40
	222222201	62	48
	222222202	62	40
	222222203	62	40
	222222204	62	40
	222222205	62	40
	333333300	63	40
	333333301	63	46
	444444400	91	40
	444444401	91	40
	444444402	91	40
	444444403	91	40
	555555500	92	40
	555555501	92	44
	666666601	91	40
	666666603	91	40
	666666604	91	40
	666666605	92	40
	666666606	91	40
	666666607	61	40
	666666608	62	40
	666666609	63	40
	666666610	61	40
	666666611	61	40
	666666612	61	40
	666666613	61	30
	666666613	62	10
	666666613	63	10

EMPLOYEE									
FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DN
James	E	Borg	888665555	10-Nov-27	450 Stone, Houston, TX	M	55000	null	
Franklin	T	Wong	333445555	8-Dec-45	638 Voss, Houston, TX	M	40000	888665555	
Jennifer	S	Wallace	987654321	20-Jun-31	291 Berry, Bellaire, TX	F	43000	888665555	
John	B	Smith	123456789	9-Jan-55	731 Fondren, Houston, TX	M	30000	333445555	
Alicia	J	Zelaya	999887777	19-Jul-58	3321 Castle, Spring, TX	F	25000	987654321	
Ramesh	K	Narayan	666884444	15-Sep-52	971 Fire Oak, Humble, TX	M	38000	333445555	
Joyce	A	English	453453453	31-Jul-62	5631 Rice, Houston, TX	F	25000	333445555	
Ahmad	V	Jabbar	987987987	29-Mar-59	980 Dallas, Houston, TX	M	25000	987654321	
Jared	D	James	111111100	10-Oct-66	123 Peachtree, Atlanta, GA	M	85000	null	
Alex	D	Freed	444444400	9-Oct-50	4333 Pillsbury, Milwaukee, WI	M	89000	null	
John	C	James	555555500	30-Jun-75	7676 Bloomington, Sacramento, (	M	81000	null	
Jon	C	Jones	111111101	14-Nov-67	111 Allgood, Atlanta, GA	M	45000	111111100	
Justin	null	Mark	111111102	12-Jan-66	2342 May, Atlanta, GA	M	40000	111111100	
Brad	C	Knight	111111103	13-Feb-68	176 Main St., Atlanta, GA	M	44000	111111100	
Evan	E	Wallis	222222200	16-Jan-58	134 Pelham, Milwaukee, WI	M	92000	null	
Josh	U	Zell	222222201	22-May-54	266 McGrady, Milwaukee, WI	M	56000	222222200	
Andy	C	Vile	222222202	21-Jun-44	1967 Jordan, Milwaukee, WI	M	53000	222222200	
Tom	G	Brand	222222203	16-Dec-66	112 Third St, Milwaukee, WI	M	62500	222222200	
Jenny	F	Vos	222222204	11-Nov-67	263 Mayberry, Milwaukee, WI	F	61000	222222201	
Chris	A	Carter	222222205	21-Mar-60	565 Jordan, Milwaukee, WI	F	43000	222222201	
Kim	C	Grace	333333300	23-Oct-70	6677 Mills Ave, Sacramento, CA	F	79000	null	
Jeff	H	Chase	333333301	7-Jan-70	145 Bradbury, Sacramento, CA	M	44000	333333300	
Bonnie	S	Bays	444444401	19-Jun-56	111 Hollow, Milwaukee, WI	F	70000	444444400	
Alec	C	Best	444444402	18-Jun-66	233 Solid, Milwaukee, WI	M	60000	444444400	
Sam	S	Snedder	444444403	31-Jul-77	987 Windy St, Milwaukee, WI	M	48000	444444400	
Nandita	K	Ball	555555501	16-Apr-69	222 Howard, Sacramento, CA	M	62000	555555500	
Bob	B	Bender	666666600	17-Apr-68	8794 Garfield, Chicago, IL	M	96000	null	
Jill	J	Jarvis	666666601	14-Jan-66	6234 Lincoln, Chicago, IL	F	36000	666666600	
Kate	W	King	666666602	16-Apr-66	1976 Boone Trace, Chicago, IL	F	44000	666666600	
Lyle	G	Leslie	666666603	9-Jun-63	417 Hancock Ave, Chicago, IL	M	41000	666666601	
Billie	J	King	666666604	1-Jan-60	556 Washington, Chicago, IL	F	38000	666666603	
Jon	A	Kramer	666666605	22-Aug-64	1988 Windy Creek, Seattle, WA	M	41500	666666603	
Ray	H	King	666666606	16-Aug-49	213 Delk Road, Seattle, WA	M	44500	666666604	
Gerald	D	Small	666666607	15-May-62	122 Ball Street, Dallas, TX	M	29000	666666602	
Arnold	A	Head	666666608	19-May-67	233 Spring St, Dallas, TX	M	33000	666666602	
Helga	C	Pataki	666666609	11-Mar-69	101 Holyoke St, Dallas, TX	F	32000	666666602	
Naveen	B	Drew	666666610	23-May-70	198 Elm St, Philadelphia, PA	M	34000	666666607	
Carl	E	Reedy	666666611	21-Jun-77	213 Ball St, Philadelphia, PA	M	32000	666666610	
Sammy	G	Hall	666666612	11-Jan-70	433 Main Street, Miami, FL	M	37000	666666611	

4.2 mysql Utility

MySQL database system provides an interactive utility, called mysql, which allows the user to enter SQL commands interactively. One can also include one or more SQL statements in a file and have them executed within mysql using the source command.

In the following discussion, we will assume that your MySQL administrator has created a database called company[4], created a MySQL user[5] called book, and has granted all rights to the company database to the book user.

The following mysql session creates the department table:

```
$ mysql -u book -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1485 to server version: 4.1.9-standard

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use company
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> source create-department.sql;
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+-------------------+
| Tables_in_company |
+-------------------+
| department        |
| foo               |
+-------------------+
2 rows in set (0.00 sec)

mysql> exit;
Bye
$
```

[4] The administrator command to create a MySQL user is: `create user book identified by 'book';` Here the user id is book and the password is also book.
[5] The administrator command to create a database called company and to assign all rights to the book user is: `grant all on company.* to 'book'@'hostname.domain.edu';`

In the above `mysql` session, the user invokes the `mysql` program with the userid `book` and the `-p` option to specify the password in a separate line. After connecting to the database server, the `use company` command is executed to start using the company database. Then, the `source` command is executed on the file `create-department.sql` that contains the following SQL statement:

```
CREATE TABLE department (
  dname        varchar(25) not null,
  dnumber      integer(4),
  mgrssn       char(9) not null,
  mgrstartdate date,
  primary key (dnumber),
  key (dname)
);
```

Note: The data types supported in different DBMSs may have slightly different names; Please consult MySQL documentation to learn about all data types supported.

Bulk Loading of Data

MySQL provides a "`load`" command that can load data stored in a text file into a table. The following `mysql` session illustrates the loading of the data located in `department.csv` file into the `department` table. In the following example, the load command is directly entered at the `mysql` prompt. It may be a good idea to create a script file with the `load` command and use the `source` command to execute the `load` command. This way, if there are any syntax errors, these can be corrected in the script file and the `load` command can be re-executed.

```
$ mysql -u raj -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1493 to server version: 4.1.9-standard

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use company;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> LOAD DATA LOCAL INFILE "department.csv"
INTO TABLE department FIELDS TERMINATED BY ","
OPTIONALLY ENCLOSED BY '"';
Query OK, 6 rows affected (0.00 sec)
Records: 6  Deleted: 0  Skipped: 0  Warnings: 0
```

```
mysql> select * from department;
+----------------+---------+-----------+--------------+
| dname          | dnumber | mgrssn    | mgrstartdate |
+----------------+---------+-----------+--------------+
| Research       |       5 | 333445555 | 1978-05-22   |
| Administration |       4 | 987654321 | 1985-01-01   |
| Headquarters   |       1 | 888665555 | 1971-06-19   |
| Software       |       6 | 111111100 | 1999-05-15   |
| Hardware       |       7 | 444444400 | 1998-05-15   |
| Sales          |       8 | 555555500 | 1997-01-01   |
+----------------+---------+-----------+--------------+
6 rows in set (0.00 sec)

mysql> exit;
Bye
$
```

The `LOAD DATA` command takes the name of the data file as parameter and other information such as field terminator symbols (in this case the comma) and loads the data into the table. For more details on the `load` command, please consult the MySQL documentation.

4.3 MySQL and PHP Programming

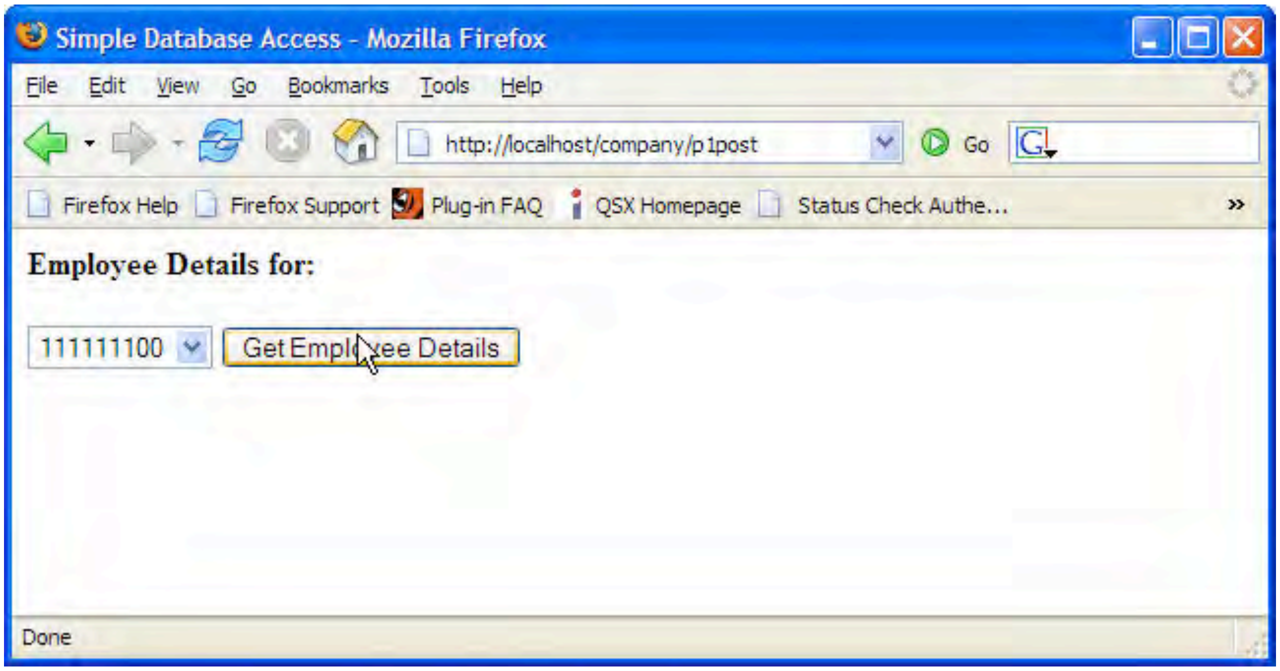
PHP is a very popular Web scripting language that allows the programmers to rapidly develop Web applications. In particular, PHP is most suited to develop Web applications that access a MySQL database. In this section, we illustrate the ease of programming with PHP and provide several examples of Web access to MySQL databases.

Example 1: Consider the problem of finding employee names given their social security number. To implement this problem as a Web application, we can design two Web pages:

1. The first Web page would contain a HTML form that contains a select list of social security numbers of employees and a submit button.

2. Upon choosing a social security number and submitting the form in the first Web page produces the second Web page that lists the name of the employee.

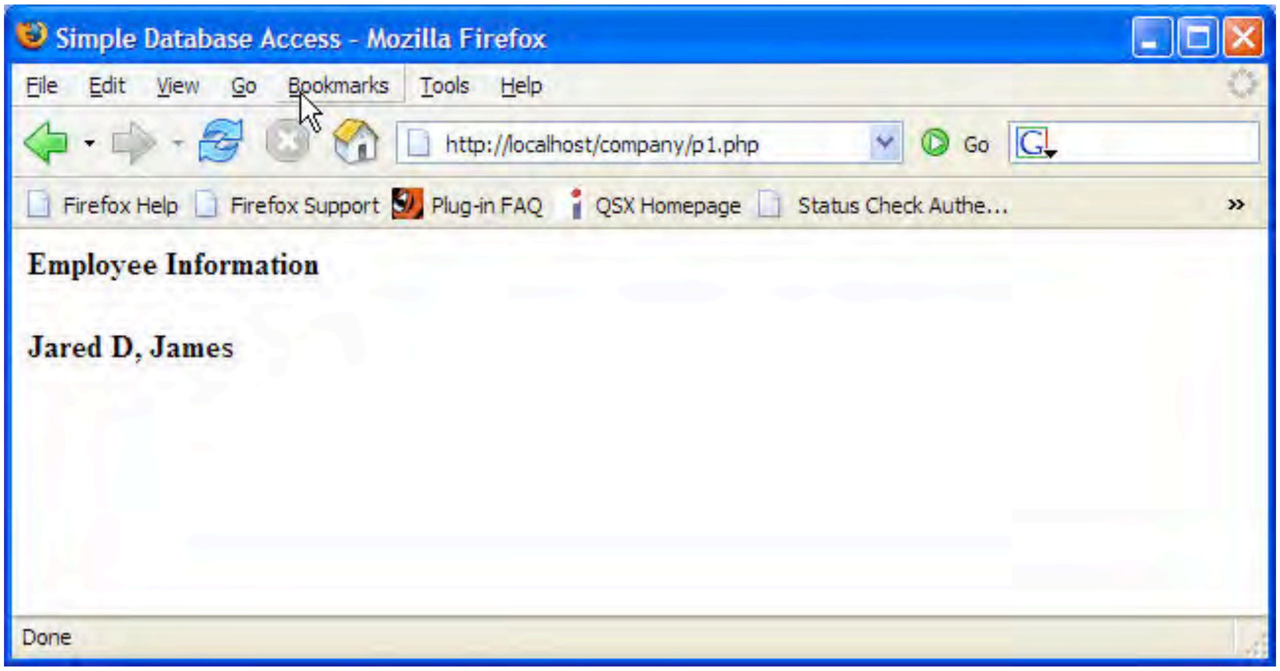
The two Web pages are shown in Figures 4.1 and 4.2.


Simple Database Access - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/company/p1post
Go
Firefox Help
Firefox Support
Plug-in FAQ
QSX Homepage
Status Check Authe...
Done

Employee Details for:

111111100 | Get Employee Details

Figure 4.1: Initial Web Page - Example 1


Simple Database Access - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/company/p1.php
Go
Firefox Help
Firefox Support
Plug-in FAQ
QSX Homepage
Status Check Authe...
Done

Employee Information

Jared D, James

Figure 4.2: Second Web Page - Example 1

Both these Web pages contain dynamic information (obtained from the database) and therefore can easily be produced by PHP scripts. The PHP script (`p1post.php`) that produces the first Web page is shown below.

```
<html>
<head>
<title>Simple Database Access</title>
</head>
<body>

<?
$username="user";
$password="password";
$database="company";
mysql_connect(localhost,$username,$password);
@mysql_select_db($database) or die( "Unable to select database");
$query="SELECT ssn FROM employee";
$result=mysql_query($query);
$num=mysql_numrows($result);
mysql_close();
?>

<h4>Employee Details for:</h4>
<form method="post" action="p1.php">
<select name="ssn">

<?
$i=0;
while ($i < $num) {
  $ssn=mysql_result($result,$i,"ssn");
  echo "<option>",$ssn,"\n";
  $i++;
}
?>

</select>
<input type="submit" value="Get Employee Details">
</form>
</body>
</html>
```

A PHP script typically consists of HTML code to display “static” parts of the Web page interspersed with procedural PHP statements that produce “dynamic” parts of the Web page. The dynamic content may come from a database such as MySQL and hence most of the PHP procedural code involves connecting to the database, running queries and using the query result to produce parts of the Web page.

In the above example script, a simple HTML page is produced which has:

- Some static content such as text headers and a HTML form with a submit button. The HTML form when submitted invokes the second PHP script called “`p1.php`”.

- A dynamic "select" GUI element within the HTML form which contains a list of employee social security numbers for the users to choose from.

PHP statements are enclosed within `<?` And `>?`. HTML code can be produced within PHP code using the "`echo`" command as is seen in several places in the code. As can be seen, there are two blocks of PHP code in the example: one to connect to the database and execute a query and the second to use the results of the query to produce the HTML "select" list options.

The PHP script (`p1.php`) to produce the second Web page is shown next.

```
<html>
<head>
<title>Simple Database Access</title>
</head>
<body>
<h4>Employee Information</h4>

<?
$username="user";
$password="password";
$database="company";
$ssn=$_POST['ssn'];
mysql_connect(localhost,$username,$password);
@mysql_select_db($database) or die( "Unable to select database");
$query="SELECT * FROM employee where ssn=$ssn";
$result=mysql_query($query);
$num=mysql_numrows($result);
mysql_close();
if ($num == 1) {
  $fname=mysql_result($result,$i,"fname");
  $minit=mysql_result($result,$i,"minit");
  $lname=mysql_result($result,$i,"lname");
  echo "<b>$fname $minit, $lname</b>";
}?>

</body>
</html>
```

In this script, the employee social security number posted in the first Web page is retrieved using the PHP statement

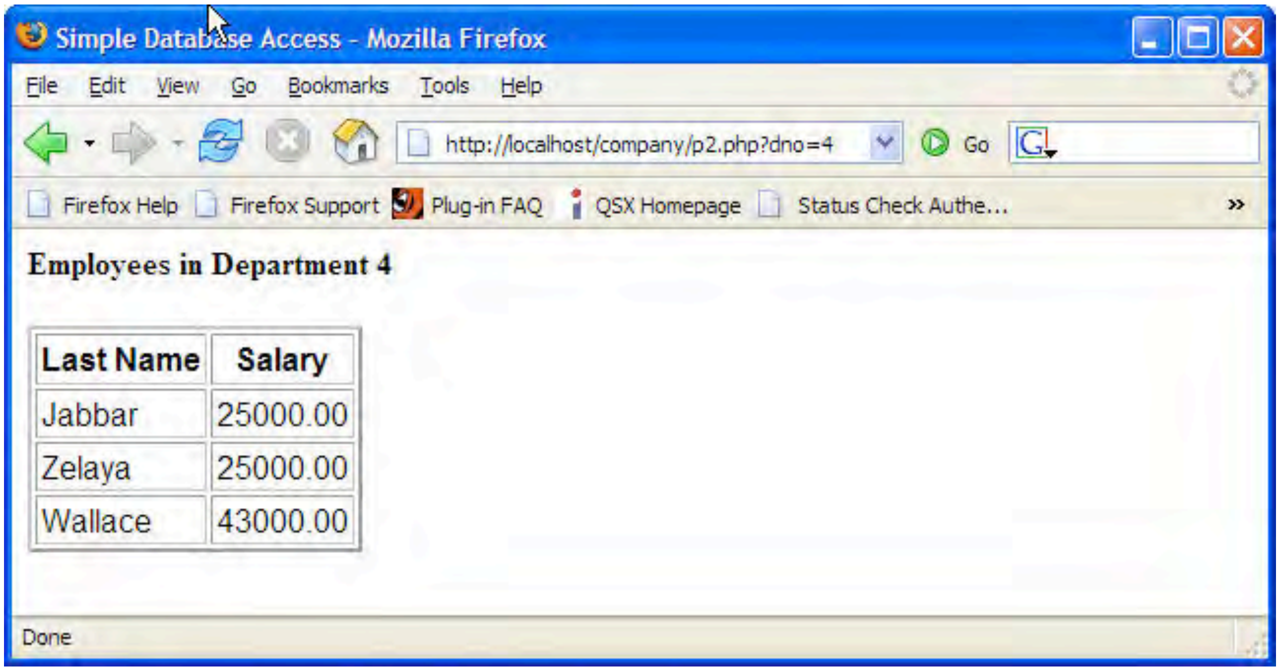
```
$ssn=$_POST['ssn'];
```

This social security number is then used in an SQL query to retrieve employee name. The script contains one block of PHP code surrounded by some HTML code.

Example 2: In this example, a PHP script that lists all employees in a given department is shown. The script takes as input the department number as a "GET" parameter in the URL itself as follows:

```
http://localhost/company/p2?dno=4
```

The Web page produced by the script is shown in Figure 4.3.


Simple Database Access - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/company/p2.php?dno=4
Go
Firefox Help
Firefox Support
Plug-in FAQ
QSX Homepage
Status Check Authe...
Done

Employees in Department 4

Last Name	Salary
Jabbar	25000.00
Zelaya	25000.00
Wallace	43000.00

Figure 4.3: Web Page for Example 2

The PHP code (`p2.php`) is shown below:

```
<html>
<head>
<title>Simple Database Access</title>
</head>
<body>

<?
$username="user";
$password="password";
$database="company";
$dno=$_GET['dno'];
mysql_connect(localhost,$username,$password);
@mysql_select_db($database) or die( "Unable to select database");
$query="SELECT lname,salary FROM employee where dno=$dno";
$result=mysql_query($query);
```

```
$num=mysql_numrows($result);
mysql_close();
?>

<table border="2" cellspacing="2" cellpadding="2">
<tr>
<th><font face="Arial,Helvetica,sans-serif">Last Name</font></th>
<th><font face="Arial,Helvetica,sans-serif">Salary</font></th>
</tr>

<?
echo "<h4>Employees in Department $dno</h4>";
$i=0;
while ($i < $num) {
  $lname=mysql_result($result,$i,"lname");
  $salary=mysql_result($result,$i,"salary");
?>

<tr>
<td><font face="Arial, Helvetica, sans-serif">
<? echo $lname; ?>
</font></td>
<td><font face="Arial, Helvetica, sans-serif">
<? echo $salary; ?>
</font></td>
</tr>

<?
  $i++;
}
?>

</table>
</body>
</html>
```

The PHP script performs an SQL query to retrieve employee names and salaries who work for the given department. This information is then formatted neatly into an HTML table for display. This example illustrates more intricate embedding of PHP code within HTML code as is seen in the "`while`" loop towards the end of the script.

Example 3: A COMPANY database browser application is shown in this example. The initial Web page in this application lists all the departments in the company. By following hyperlinks, the user may see more details of departments, employees, and projects in three separate Web pages. The browser program is implemented using four PHP scripts:

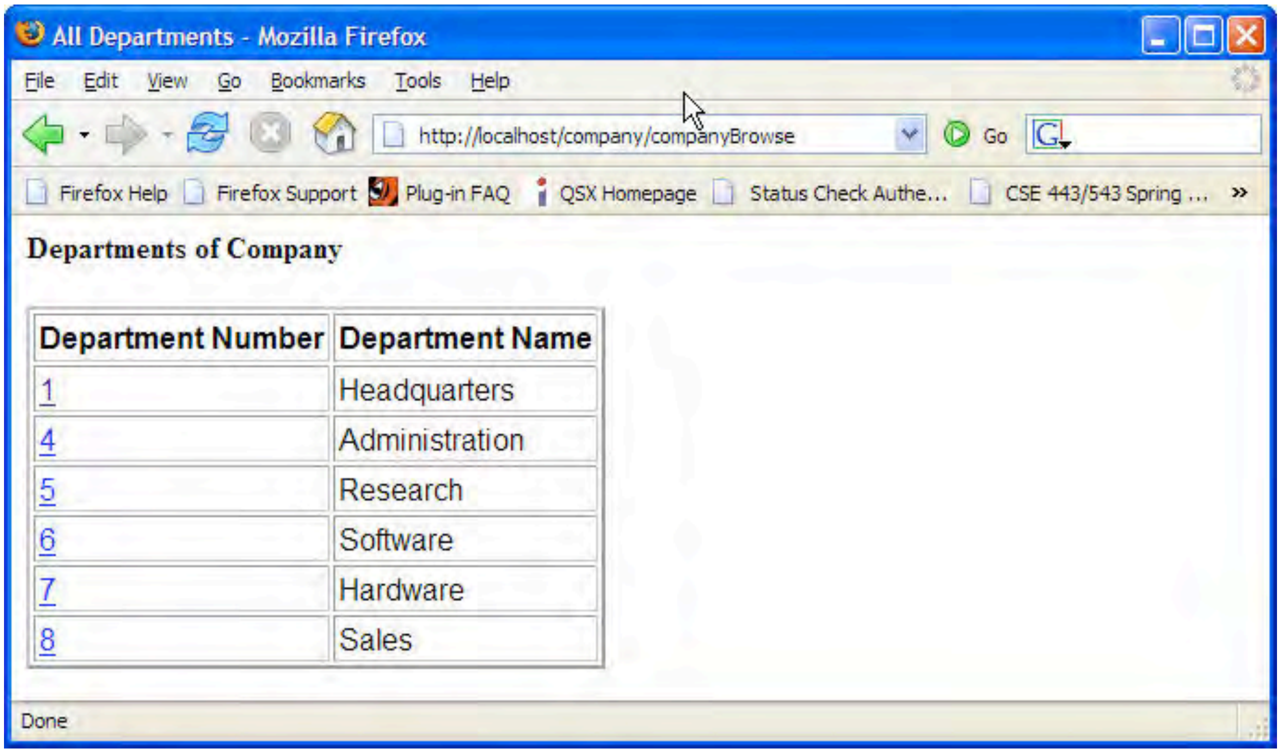
(a) `companyBrowse.php`: This script lists all the departments in the company in a tabular form as shown in Figure

(b) `deptView.php`:
(c) `empView.php`:
(d) `projectView.php`:

The code for `companyBrowse` script is shown below:

```
<html>
<head>
<title>All Departments</title>
</head>
<body>

<?
$username="user";
$password="password";
$database="company";

mysql_connect("host.domain.edu",$username,$password);
@mysql_select_db($database) or die( "Unable to select database");
$query="SELECT dnumber,dname FROM department order by dnumber";
$result=mysql_query($query);
$num=mysql_numrows($result);
mysql_close();
?>

<h4>Departments of Company</h4>
<table border="2" cellspacing="2" cellpadding="2">
<tr>
<th><font face="Arial, Helvetica, sans-serif">
    Department Number</font></th>
<th><font face="Arial, Helvetica, sans-serif">
    Department Name</font></th>
</tr>

<?
$i=0;
while ($i < $num) {
  $dno=mysql_result($result,$i,"dnumber");
  $dname=mysql_result($result,$i,"dname");
?>

<tr>
<td><font face="Arial, Helvetica, sans-serif">
  <a href="deptView?dno=<? echo $dno; ?>">
  <? echo $dno; ?></a></font></td>
<td><font face="Arial, Helvetica, sans-serif">
  <? echo $dname; ?></font></td>
</tr>

<?
  $i++;
}
?>

</table>
</body>
</html>
```

The script performs a simple query on the `DEPARTMENT` table and outputs the list of department numbers and names formatted as an HTML table as shown in Figure 4.4.

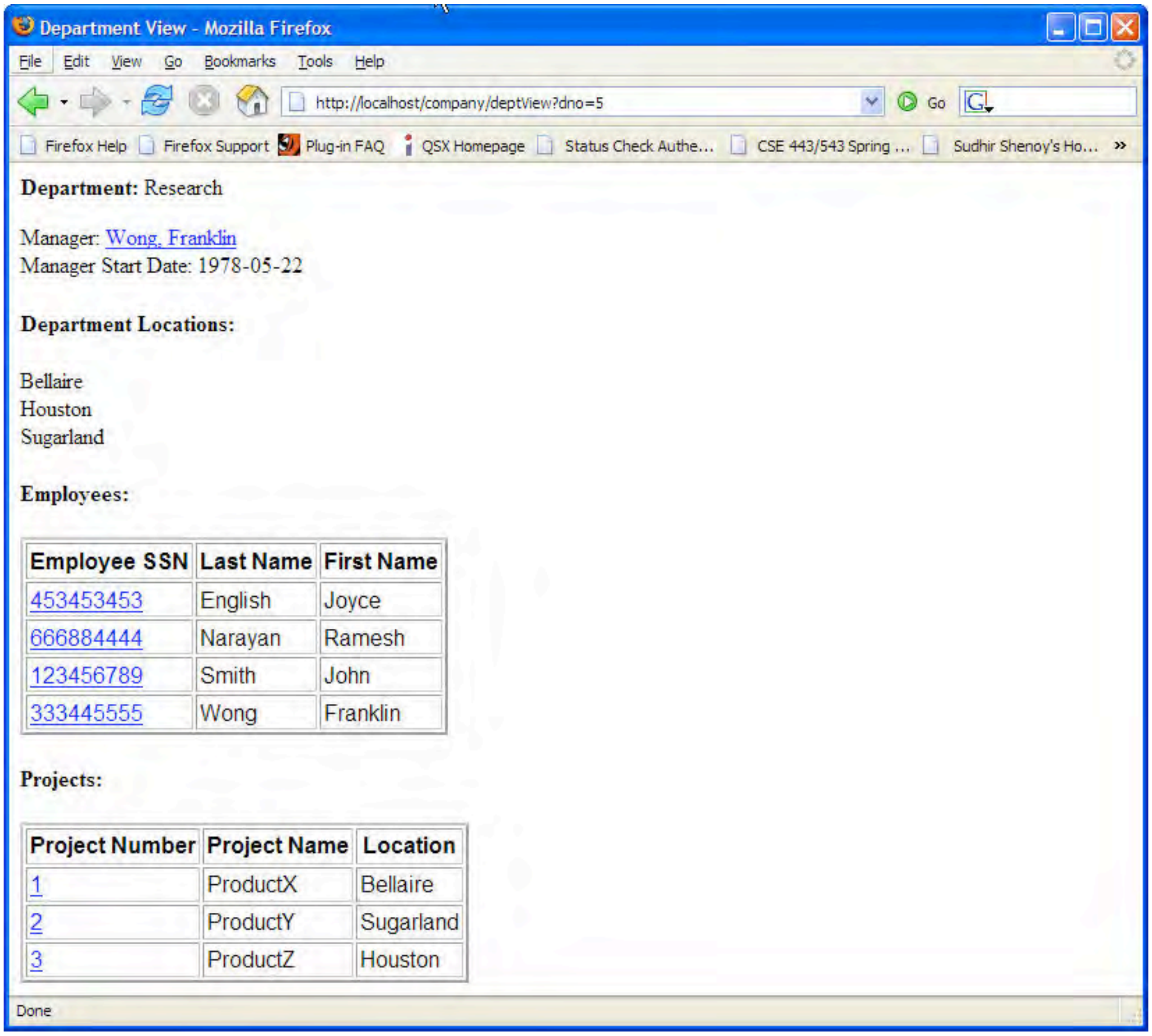

All Departments - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/company/companyBrowse
Go
Firefox Help
Firefox Support
Plug-in FAQ
QSX Homepage
Status Check Authe...
CSE 443/543 Spring ...
Done

Departments of Company

Department Number	Department Name
1	Headquarters
4	Administration
5	Research
6	Software
7	Hardware
8	Sales

Figure 4.4: All Departments Web Page – COMPANY database browser

The department numbers in this list are formatted as HTML hyperlinks that, when traversed by the user, will produce a Web page containing more details of the chosen department. The detailed department view Web page is shown in Figure 4.5.


Department View - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/company/deptView?dno=5
Go
Firefox Help Firefox Support Plug-in FAQ QSX Homepage Status Check Authe... CSE 443/543 Spring ... Sudhir Shenoy's Ho...
Done

Department: Research

Manager: Wong, Franklin
Manager Start Date: 1978-05-22

Department Locations:

Bellaire
Houston
Sugarland

Employees:

Employee SSN	Last Name	First Name
453453453	English	Joyce
666884444	Narayan	Ramesh
123456789	Smith	John
333445555	Wong	Franklin

Projects:

Project Number	Project Name	Location
1	ProductX	Bellaire
2	ProductY	Sugarland
3	ProductZ	Houston

Figure 4.5: Department Detail Web Page – COMPANY database browser

The PHP script (`deptView.php`) that produces the detailed department view is shown below.

```
<html>
<head>
<title>Department View</title>
</head>
<body>

<?
$username="user";
$password="password";
$database="company";
```

```
$dno=$_GET['dno'];
mysql_connect(“host.domain.edu”,$username,$password);
@mysql_select_db($database) or die( "Unable to select database");

$query="SELECT dname,mgrssn,mgrstartdate,lname,fname FROM
department,employee where dnumber=$dno and mgrssn=ssn";
$result=mysql_query($query);
$num=mysql_numrows($result);

$dname=mysql_result($result,0,"dname");
$mssn=mysql_result($result,0,"mgrssn");
$mstart=mysql_result($result,0,"mgrstartdate");
$mlname=mysql_result($result,0,"lname");
$mfname=mysql_result($result,0,"fname");

echo "<b>Department: </b>", $dname;
echo "<P>Manager: <a href=\"empView?", $mssn, "\">", $mlname, ",
", $mfname, "</a></BR>";
echo "Manager Start Date: ", $mstart;

echo "<h4>Department Locations:</h4>";
$query="SELECT dlocation FROM dept_locations where dnumber=$dno";
$result=mysql_query($query);
$num=mysql_numrows($result);
$i=0;
while ($i < $num) {
  $dloc=mysql_result($result,$i,"dlocation");
  echo $dloc, "<BR>\n";
  $i++;
}

echo "<h4>Employees:</h4>";
$query="SELECT ssn,lname,fname FROM employee where dno=$dno";
$result=mysql_query($query);
$num=mysql_numrows($result);
?>

<table border="2" cellspacing="2" cellpadding="2">
<tr>
<th><font face="Arial, Helvetica, sans-serif">
     Employee SSN</font></th>
<th><font face="Arial, Helvetica, sans-serif">
     Last Name</font></th>
<th><font face="Arial, Helvetica, sans-serif">
     First Name</font></th>
</tr>
```

```
<?
$i=0;
while ($i < $num) {
  $essn=mysql_result($result,$i,"ssn");
  $elname=mysql_result($result,$i,"lname");
  $efname=mysql_result($result,$i,"fname");
?>
<tr>
  <td><font face="Arial, Helvetica, sans-serif">
  <a href="empView?ssn=<? echo $essn; ?>">
     <? echo $essn; ?></a></font></td>
<td><font face="Arial, Helvetica, sans-serif">
     <? echo $elname; ?></font></td>
<td><font face="Arial, Helvetica, sans-serif">
     <? echo $efname; ?></font></td>
</tr>
<?
  $i++;
}
?>

</table>

<?
echo "<h4>Projects:</h4>";
$query="SELECT pnumber,pname,plocation FROM project where
dnum=$dno";
$result=mysql_query($query);
$num=mysql_numrows($result);
?>

<table border="2" cellspacing="2" cellpadding="2">
<tr>
<th><font face="Arial, Helvetica, sans-serif">Project
Number</font></th>
<th><font face="Arial, Helvetica, sans-serif">Project
Name</font></th>
<th><font face="Arial, Helvetica, sans-serif">Location</font></th>
</tr>

<?
$i=0;
while ($i < $num) {
  $pnum=mysql_result($result,$i,"pnumber");
  $pname=mysql_result($result,$i,"pname");
  $ploc=mysql_result($result,$i,"plocation");
?>
<tr>
  <td><font face="Arial, Helvetica, sans-serif">
  <a href="projView?ssn=<? echo $pnum; ?>"><? echo $pnum;
?></a></font></td>
<td><font face="Arial, Helvetica, sans-serif"><? echo $pname;
?></font></td>
<td><font face="Arial, Helvetica, sans-serif"><? echo $ploc;
?></font></td>
</tr>
<?
  $i++;
}

mysql_close();
?>

</body>
</html>
```

The `deptView` script executes the following four queries and formats the results of the queries as shown in the Web page.

```
SELECT dname,mgrssn,mgrstartdate,lname,fname
FROM   department,employee
WHERE  dnumber=$dno and mgrssn=ssn

SELECT dlocation
FROM   dept_locations
WHERE  dnumber=$dno

SELECT ssn,lname,fname
FROM   employee
WHERE  dno=$dno

SELECT pnumber,pname,plocation
FROM   project
WHERE  dnum=$dno
```

Each of these queries uses the PHP variable `$dno` containing the department number for which the view is generated.

The department view also contains hyperlinks for employees and projects which when traversed by the user produces detailed employee and project Web pages. The code for PHP scripts that produce these Web pages is omitted as they are similar to the `deptView` script.

4.4 Online Address Book

In this section an online address/contact book application is illustrated. The application is coded in PhP and accesses a MySQL database which has the following table:

```
create table contacts (
  lname  varchar(30),
  fname  varchar(30),
  email  varchar(30),
  homePhone varchar(20),
  cellPhone varchar(20),
  officePhone varchar(20),
  address varchar(100),
  comment varchar(100),
  primary key (lname,fname)
);
```

The application is capable of the following functions:

(1) ADD a new contact.
(2) DELETE one or more contacts.
(3) SEARCH contacts by substring match on name.
(4) LIST all contacts.

The initial Web page is shown in Figure 4.6. On the left frame, the menu options are listed and the right frame contains space to display results.

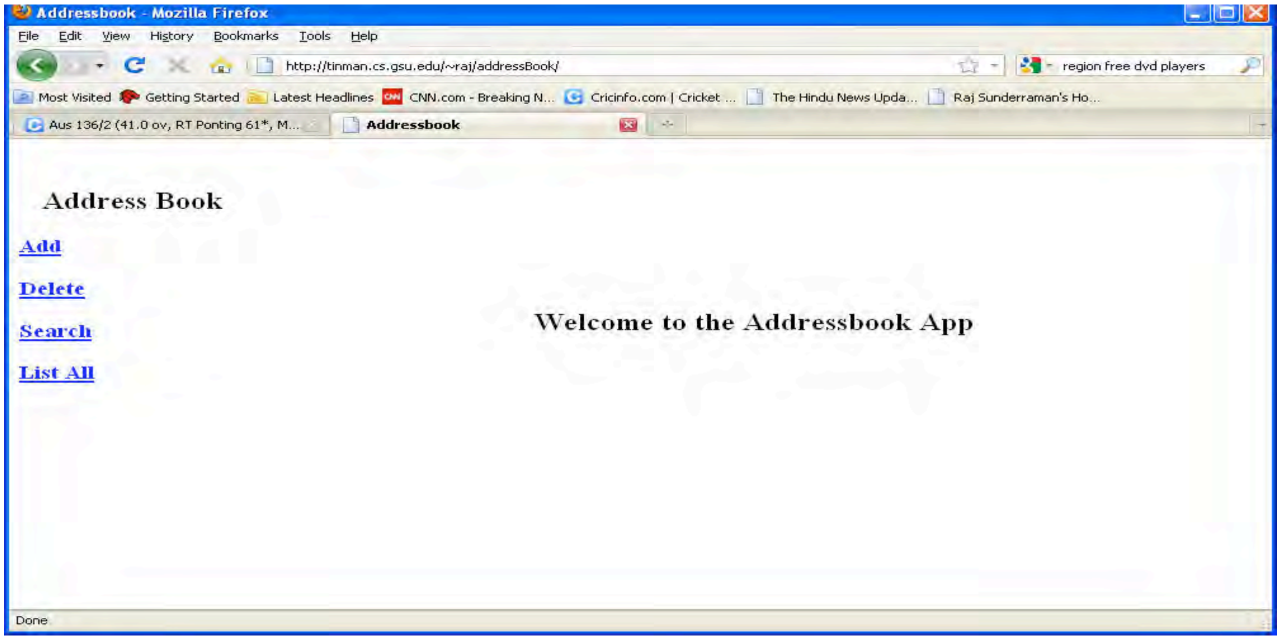

Addressbook - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://tinman.cs.gsu.edu/~raj/addressBook/
Address Book
Add
Delete
Search
List All
Welcome to the Addressbook App
Done

Figure 4.6: Address Book – Main Page

The HTML code that produces this (`index.html`) page is shown below:

```
<html>
<head>
<title>Addressbook</title>
</head>
<frameset rows="*" cols="18%,*" framespacing="0"
          frameborder="no" border="0">
  <frame src="mainMenu.html" name="leftFrame" scrolling="No"
         noresize="noresize" id="leftFrame" />
  <frame src="Welcome.html" name="mainFrame" id="mainFrame" />
</frameset>
</html>
```

The HTML code that produces the left frame (`mainMenu.html`) is shown below:

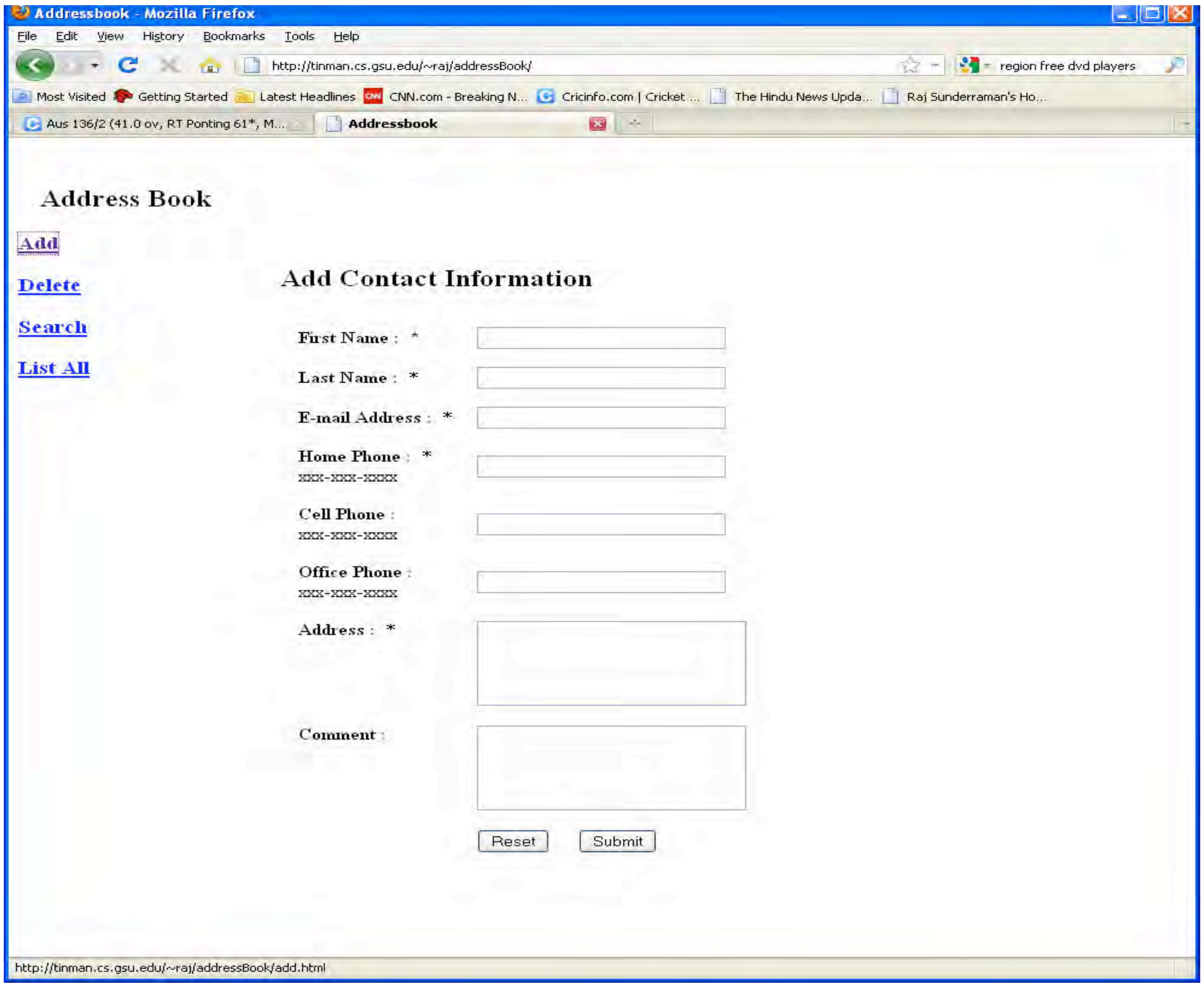
```
<html>
<head>
<title>Addressbook</title>
</head>
<body>
  <p>
  <h2>&nbsp;&nbsp;&nbsp;Address Book</h2>
  <h3><a href="add.html" target="mainFrame">Add</a></h3>
  <h3><a href="delete.php" target="mainFrame">Delete</a></h3>
  <h3><a href="search.html" target="mainFrame">Search</a> </h3>
  <h3><a href="list.php" target="mainFrame">List All</a></h3>
</body>
</html>
```

The HTML code (`Welcome.html`) to produce the right frame is shown below:

```
<html>
  <head><title>Addressbook</title></head>
  <body>
    <BR><BR><BR><BR><BR><BR><BR>
    <center>
    <h2>Welcome to the Addressbook App</h2>
    </center>
  </body>
</html>
```

Add

Upon clicking the "Add" menu option, the form shown in Figure 4.7 is displayed on the right frame:


Addressbook - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://tinman.cs.gsu.edu/~raj/addressBook/
region free dvd players
Most Visited
Getting Started
Latest Headlines
CNN.com - Breaking N...
Cricinfo.com | Cricket ...
The Hindu News Upda...
Raj Sunderraman's Ho...
Aus 136/2 (41.0 ov, RT Ponting 61*, M...
Addressbook
Reset
Submit
http://tinman.cs.gsu.edu/~raj/addressBook/add.html

Address Book

Add

Delete

Search

List All

Add Contact Information

First Name : *	
Last Name : *	
E-mail Address : *	
Home Phone : * xxx-xxx-xxxx	
Cell Phone : xxx-xxx-xxxx	
Office Phone : xxx-xxx-xxxx	
Address : *	
Comment :	

Figure 4.7: Address Book – Add Form

Mandatory fields are marked with an "*". The HTML code (`add.html`) that produces this Web page is shown below:

```
<html>
<head>
<title>Add</title>
</head>
<body>
<h2>Add Contact Information</h2>
<form name="addform" action="add.php" method=post" >
   <table width="60%" border="0" cellpadding="3" cellspacing="12">
   <tr>
      <td width="130"><strong>First Name:</strong>*</td>
      <td><input name="fname" type="text" size="30" maxlength="30" /></td>
   </tr>
   <tr>
      <td width="130"><strong>Last Name:</strong>*</td>
      <td><input name="lname" type="text" size="30" maxlength="30" /></td>
    </tr>
    <tr>
      <td width="130"><strong>E-mail Address:</strong>*</td>
      <td><input name="email" type="text" size="30" maxlength="30" /></td>
    </tr>
    <tr>
      <td width="130"><p><strong>Home Phone:</strong>*<br />xxx-xxx-xxxx</p>
      </td>
      <td><input name="homePhone" type="text" size="30" maxlength="20" /></td>
    </tr>
    <tr>
      <td width="130"><strong>Cell Phone : <br />
      </strong>xxx-xxx-xxxx</td>
      <td><input name="cellPhone" type="text" size="30" maxlength="20" /></td>
    </tr>
    <tr>
      <td width="130"><strong>Office Phone : <br />
      </strong>xxx-xxx-xxxx</td>
      <td><input name="officePhone" type="text" size="30"
                 maxlength="20" /></td>
    </tr>
    <tr valign="top">
      <td width="130"><strong>Address:</strong>*</td>
      <td><textarea name="address" cols="25" rows="4"></textarea></td>

    </tr>
    <tr valign="top">
      <td width="130"><strong>Comment:</strong> </td>
      <td><textarea name="comment" cols="25" rows="4"></textarea></td>
    </tr>
    <tr>
      <td width="130">&nbsp;</td>
      <td><input type="reset" name="Reset" value="Reset" />
        &nbsp;&nbsp;&nbsp;&nbsp;
        <input type="submit" name="Submit" value="Submit" /></td>
    </tr>
  </table>
  </form>
</body>
</html>
```

The code to process the submission of the "ADD" form (`add.php`) is shown below:

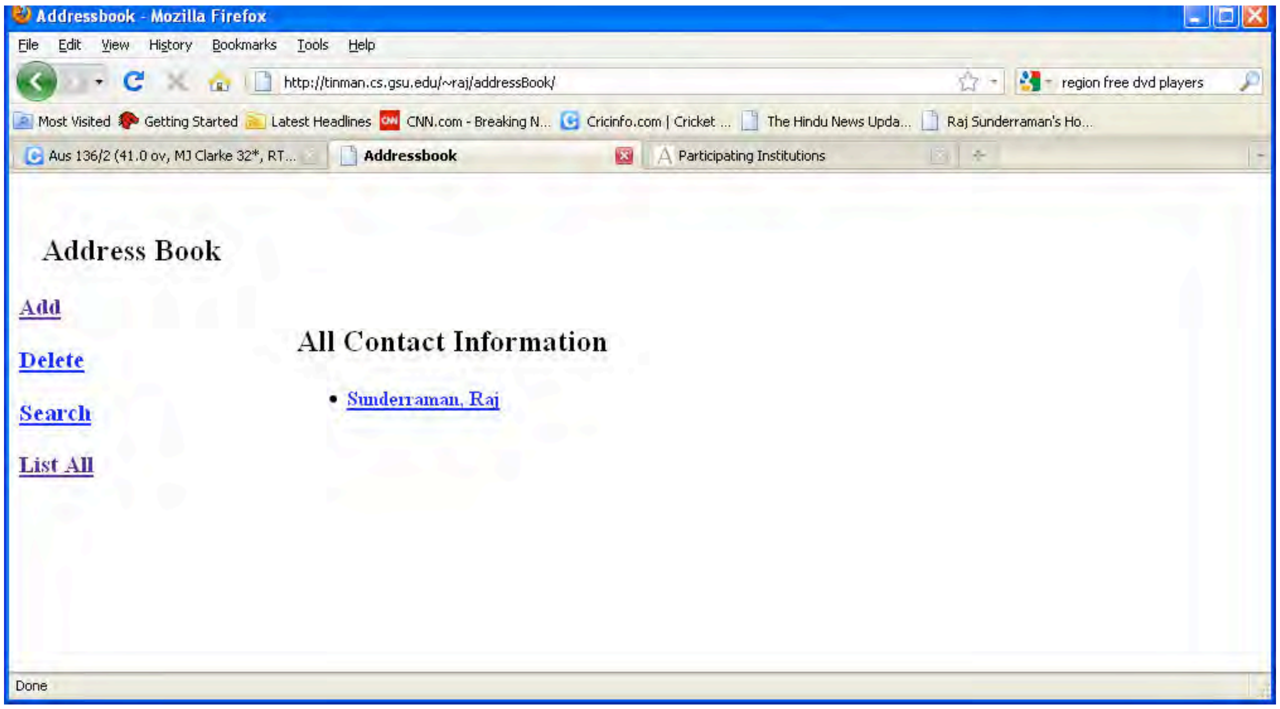
```
<?php
// connect to my sql
$host="localhost"; // Host name
$username="id";        // Mysql username
$password="pwd";     // Mysql password
$db_name="db";        // Database name
$tbl_name="contacts"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// retrieve all variables
$fname = @$_POST["fname"];
```

```
$lname = @$_POST["lname"];
$email = @$_POST["email"];
$homePhone = @$_POST["homePhone"];
$cellPhone = @$_POST["cellPhone"];
$officePhone = @$_POST["officePhone"];
$address = @$_POST["address"];
$comment = @$_POST["comment"];

// insert information to database
$sql="insert into $tbl_name values”.
     "('$lname','$fname','$email','$homePhone','$cellPhone', ".
     "'$officePhone','$address','$comment')";
$result = mysql_query($sql);
mysql_close();
?>

<html>
<head>
<title>Add processed</title>
<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<blockquote>
  <p>
  <h3>Your information is added to database. </h3>
  <body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
</body>
</html>
```

The program first connects to the database, then retrieves all submitted data into variables and then constructs an SQL insert statement. Finally, it executes the insert statement and prints a message.

List All

Upon clicking the “List ALL” option, the Web page shown in Figure 4.8 is shown on the right frame. This is an alphabetical listing of all contacts, each hyper-linked to go to a detail page.



Addressbook - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://tinman.cs.gsu.edu/~raj/addressBook/
Addressbook
Done

Address Book

Add

Delete

Search

List All

All Contact Information

- Sunderraman, Raj

Figure 4.8: Address Book – List All

The code to display the list is shown below:

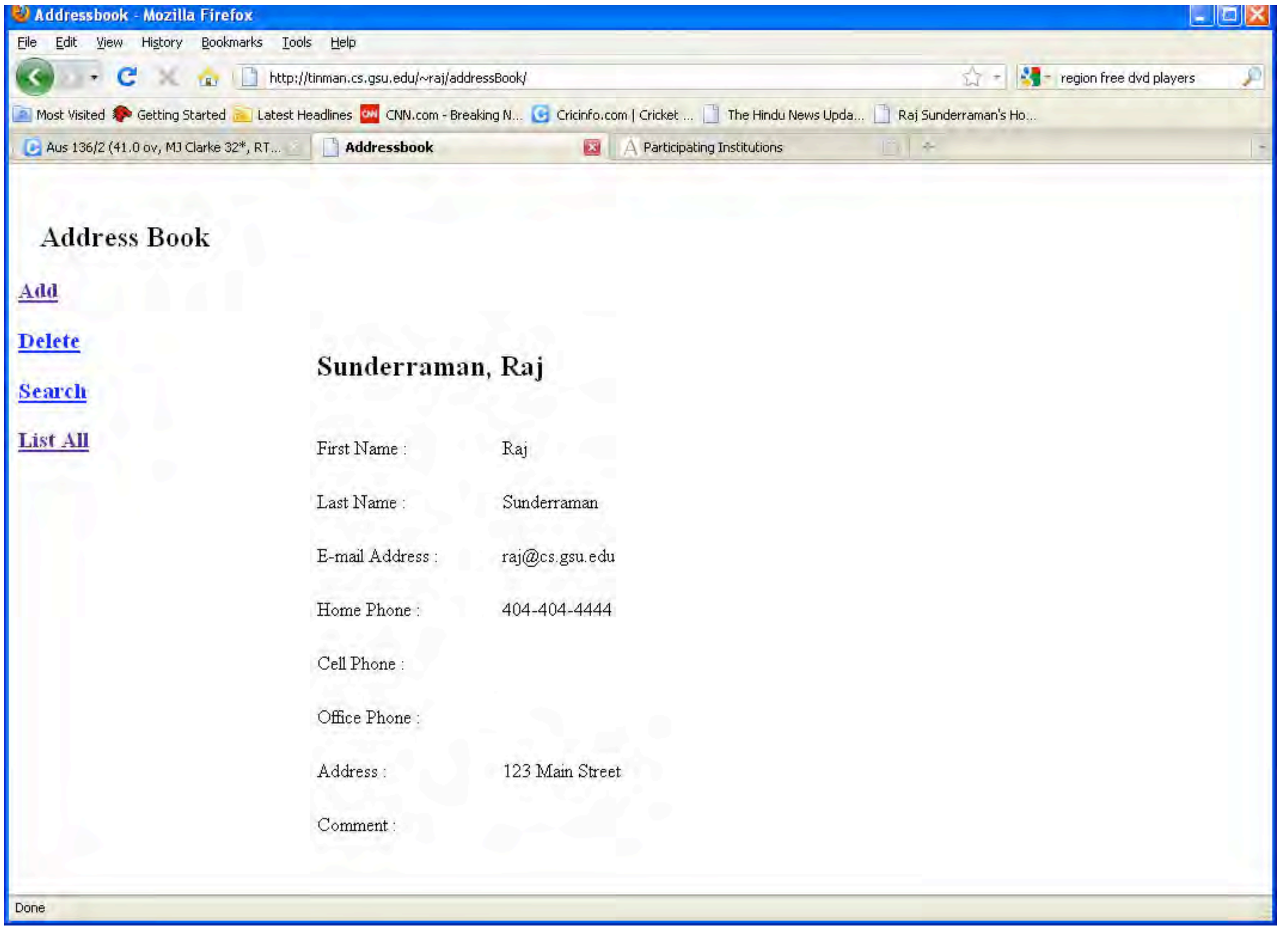
```
<?php
// connect to my sql
$host="localhost"; // Host name
$username="id";  // Mysql username
$password="pwd";  // Mysql password
$db_name="db";  // Database name
$tbl_name="contacts"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// show all contact information
$sql="select * from $tbl_name order by lname";
$result = mysql_query($sql);
mysql_close();
?>

<html>
<head>
<title>List</title>
</head>
<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<blockquote>
  <p>
  <h2>All Contact Information</h2>
```

```
   <?php
       if (mysql_num_rows($result)==0){
      echo "<h4>No data<h4>";
     } else {
       while($row = mysql_fetch_assoc($result)) {
         $lname = $row['lname'];
         $fname = $row['fname'];
         echo "<ul><li><h4><a href=\"detail.php? ".
              "lname=$lname&fname=$fname\">$lname, $fname</a><h4></li></ul>";
      }
    }
  ?>
</blockquote>
</body>
</html>
```

The program connects to the database and executes a simple query and displays the results on the Web page. Each contact is hyper-linked to the `detail.php` program.

Detail

Upon clicking the link, the detail page shown in Figure 4.9 is shown.


Addressbook - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://tinman.cs.gsu.edu/~raj/addressBook/
region free dvd players
Most Visited
Getting Started
Latest Headlines
CNN.com - Breaking N...
Cricinfo.com | Cricket ...
The Hindu News Upda...
Raj Sunderraman's Ho...
Aus 136/2 (41.0 ov, MJ Clarke 32*, RT...
Addressbook
Participating Institutions
Done

Address Book

Add

Delete

Search

List All

Sunderraman, Raj

First Name :	Raj
Last Name :	Sunderraman
E-mail Address :	raj@cs.gsu.edu
Home Phone :	404-404-4444
Cell Phone :	
Office Phone :	
Address :	123 Main Street
Comment :	

Figure 4.9: Address Book – Detail

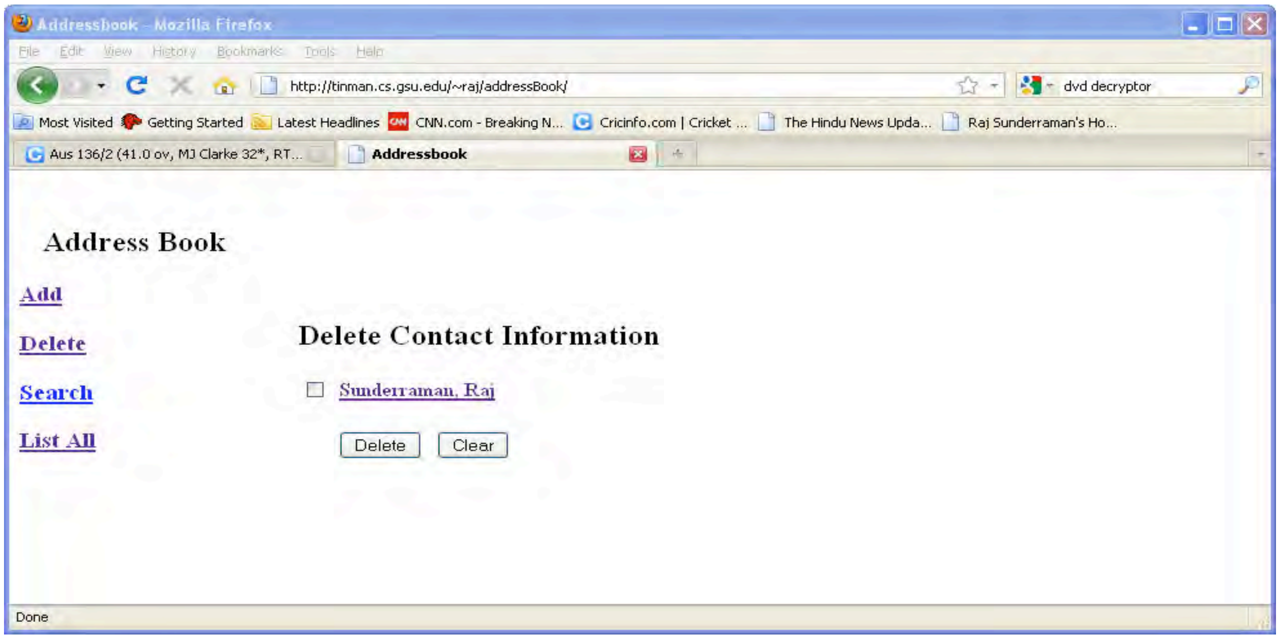
The code that produces the detail page is shown below:

```
<?php
// connect to my sql
$host="tinman.cs.gsu.edu"; // Host name
$username="cms";         // Mysql username
$password="cms123";      // Mysql password
$db_name="conf";         // Database name
$tbl_name="contacts"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// retrieve all variables
$fname = @$_GET["fname"];
$lname = @$_GET["lname"];

// show all contact information
$sql="select * from $tbl_name where fname='$fname' and lname='$lname'";
$result = mysql_query($sql);
$row = mysql_fetch_assoc($result);
mysql_close();
?>

<html>
<head>
<title>Detail</title>
</head>
<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<blockquote>
    <table width="60%" border="0" cellpadding="5" cellspacing="15">
        <tr>
            <td colspan="2"><p><h2><?php echo "$lname, $fname"; ?></h2></td>
          </tr>
    <tr>
      <td width="130">First Name :</td>
      <td><?php echo $row['fname']; ?></td>
    </tr>
    <tr>
      <td width="130">Last Name :</td>
      <td><?php echo $row['lname']; ?></td>
    </tr>
    <tr>
      <td width="130">E-mail Address :</td>
      <td><?php echo $row['email']; ?></td>
    </tr>
    <tr>
      <td width="130"><p>Home Phone :<br />
      </p>        </td>
      <td><?php echo $row['homePhone']; ?></td>
    </tr>
    <tr>
      <td width="130">Cell Phone :</td>
      <td><?php echo $row['cellPhone']; ?></td>
    </tr>
    <tr>
      <td width="130">Office Phone :</td>
      <td><?php echo $row['officePhone']; ?></td>
    </tr>
    <tr valign="top">
      <td width="130">Address :</td>
      <td><?php echo $row['address']; ?></td>
    </tr>
    <tr valign="top">
      <td width="130">Comment :</td>
      <td><?php echo $row['comment']; ?></td>
    </tr>
  </table>
   </p>
</blockquote>
</body>
</html>
```

The above code connects to the database and performs a query to get details of the requested contact. It then displays the data for the contact in a tabular format.

Delete

The delete contact interface is shown in Figure 4.10. The list of contacts is presented to the user along with a check box to the left of each contact. The user may choose one or more contacts and submit for deletion.


Addressbook - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://tinman.cs.gsu.edu/~raj/addressBook/
dvd decryptor
Most Visited
Getting Started
Latest Headlines
CNN.com - Breaking N...
Cricinfo.com | Cricket ...
The Hindu News Upda...
Raj Sunderraman's Ho...
Aus 136/2 (41.0 ov, MJ Clarke 32*, RT...
Addressbook
Done

Address Book

Add

Delete

Search

List All

Delete Contact Information

☐ Sunderraman, Raj

Delete Clear

Figure 4.10: Address Book – Delete

The code to display the delete interface as well as to process the delete request (`delete.php`) is shown below:

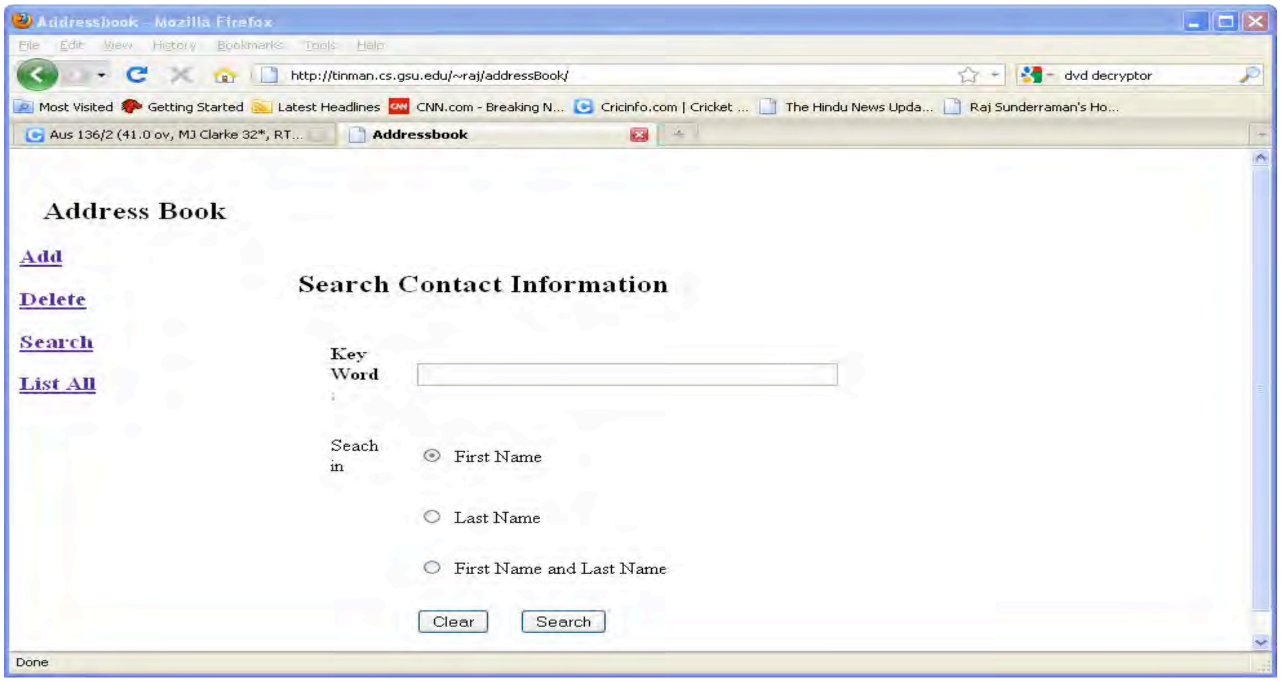
```
<?php
// connect to my sql
$host="localhost"; // Host name
$username="id";  // Mysql username
$password="pwd";  // Mysql password
$db_name="db";  // Database name
$tbl_name="contacts"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// delete record
$delete = @$_POST["delete"];
if ($delete == "Delete") {
  $dfname = @$_POST["dfname"];
  $dlname = @$_POST["dlname"];
  $checkbox = @$_POST["checkbox"];
  // delete record
  foreach ($checkbox as $index){
    $sql = "delete from $tbl_name where fname='$dfname[$index]' ".
           "and lname='$dlname[$index]'";
    $result = mysql_query($sql);
  }
}

// show all contact information
$sql="select * from $tbl_name order by lname";
$result = mysql_query($sql);
mysql_close();
?>

<html
<head>
<title>Delete</title>
</head>
<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<blockquote>
<p>
<h2>Delete Contact Information</h2>
<form name="myform" method="post" action="delete.php">
<table width="40%" border="0">
<?php
  $index=0;
  while($row = mysql_fetch_assoc($result)) {
    $lname = $row['lname'];
    $fname = $row['fname'];
    echo "<tr><td width=\"25\" valign=\"top\">".
         "<input type=\"checkbox\" name=\"checkbox[]\" value=\"$index\" />";
    echo "</td><td valign=\"bottom\"><h4><a href=\"detail.php? ".
```

```
        "lname=$lname&fname=$fname\">$lname, $fname</a><h4></td></tr>";
     echo "<input type=\"hidden\" name=\"dfname[]\" value=\"$fname\" />";
     echo "<input type=\"hidden\" name=\"dlname[]\" value=\"$lname\" />";
     $index++;
   }

   echo "<tr><td width=\"25\" valign=\"top\">&nbsp;</td>".
        "<td valign=\"bottom\"><input type=\"submit\" name=\"delete\"";
   echo "value=\"Delete\" />&nbsp;&nbsp;&nbsp; ".
        "<input type=\"reset\" name=\"Submit2\" value=\"Clear\" /></td></tr>";
?>
</table>
</form>

<?php
  if (mysql_num_rows($result)==0)
    echo "<h4>No data<h4>";
?>
</p>
</blockquote>
</body>
</html>
```

In the first part of the code, the database connection is established and if the delete submission is detected, the corresponding record is deleted from the database. Then, the listing of all contacts is produced along with the check boxes within an HTML form.

Search

The search interface is shown in Figure 4.11.



Addressbook - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://tinman.cs.gsu.edu/~raj/addressBook/
dvd decryptor
Most Visited Getting Started Latest Headlines CNN.com - Breaking N... Cricinfo.com | Cricket ... The Hindu News Upda... Raj Sunderraman's Ho...
Aus 136/2 (41.0 ov, MJ Clarke 32*, RT...
Addressbook
Address Book
Add
Delete
Search
List All
Done

Search Contact Information

Key Word :

Seach in [✓ First Name] Last Name First Name and Last Name

Clear Search

Figure 4.11: Address Book – Search

The search interface consists of a keyword text box along with options for choosing first name, last name or both for the search. The code (`search.html`) to produce the interface is shown below:

```
<html>
<head>
<title>Search HTML</title>
</head>
<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<blockquote>
<p>
<h2>Search Contact Information</h2>
<form action="result.php" method=post>
  <table width="60%" border="0" cellpadding="5" cellspacing="20">
  <tr>
    <td width="115"><strong>Key Word :  </strong></td>
    <td><input name="keyword" type="text" size="50" maxlength="50" /></td>
  </tr>
  <tr>
    <td width="115">Seach in  </td>
    <td><input name="searchin" type="radio" value="fname" checked/>
        &nbsp;First Name </td>
  </tr>
  <tr>
    <td width="115">&nbsp;</td>
    <td><input name="searchin" type="radio" value="lname" />
        &nbsp;Last Name </td>
  </tr>
  <tr>
    <td width="115">&nbsp;</td>
    <td><input name="searchin" type="radio" value="both" />
    &nbsp;First Name and Last Name </td>
  </tr>

  <tr>
    <td>&nbsp;</td>
    <td><input type="reset" name="Reset" value="Clear" />
        &nbsp;&nbsp;&nbsp;&nbsp;
        <input type="submit" name="Submit" value="Search" /></td>
  </tr>
  </table>
</form>
</p>
</blockquote>
</body>
</html>
```

The above is a straightforward coding of an HTML form with the relevant form elements enclosed within it. Upon submission of the search form, the following program (`result.php`) is called.

```
<?php
// connect to mysql
$host="localhost"; // Host name
```

```
$username="id";  // Mysql username
$password="pwd";  // Mysql password
$db_name="db";  // Database name
$tbl_name="contacts"; // Table name

// Connect to server and select database.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// retrieve all variables
$keyword = @$_POST["keyword"];
$searchin = @$_POST["searchin"];

// execute query
$sql="select * from $tbl_name";
$result = mysql_query($sql);
mysql_close();
?>

<html>
<head>
<title>Results</title>
</head>
<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<blockquote>
<p>
<h2>Result</h2>
<?php
  $i =0;
  while($row = mysql_fetch_assoc($result)) {
    $lname = $row['lname'];
    $fname = $row['fname'];
    if ($searchin == "both"){
      // search in last name & first name
      if ((preg_match("/$keyword/i",$fname))||
          (preg_match('/$keyword/',$lname))){
        echo "<ul><li><h4><a href=\"detail.php? ".
             "lname=$lname&fname=$fname\">$lname, $fname</a><h4></li></ul>";
        $i++;
      }
    } else if($searchin == "fname"){
      // search in first name
      if (preg_match("/$keyword/i",$fname)){
        echo "<ul><li><h4><a href=\"detail.php?.
             "lname=$lname&fname=$fname\">$lname, $fname</a><h4></li></ul>";
        $i++;
      }
    } else {
      // search in last name
      if (preg_match("/$keyword/i",$lname)){
        echo "<ul><li><h4><a href=\"detail.php? ".
             "lname=$lname&fname=$fname\">$lname, $fname</a><h4></li></ul>";
        $i++;
      }
      }
    }
    if ($i == 0)
      echo "<ul><h4>No match result.<h4></ul>";
?>
</blockquote>
</body>
</html>
```

The above code connects to the database and executes a query to retrieve all contacts. Then, for each contact the appropriate filter is applied and only those contacts that pass the filter test are displayed in the Web page.

Exercises

NOTE: All references to Exercises and Laboratory Exercises in the following problems refer to the numbering in the Elmasri/Navathe text.

1. Consider the ER-schema generated for the `CONFERENCE_REVIEW` database in Laboratory Exercise 7.34 for which the relational database was created in Laboratory Exercise 9.13.
   a. Create the database tables in MySQL.
   b. Create comma separated data files containing data for at least 10 papers, 15 authors, and 8 reviewers. You must assign authors to papers, assign a contact author, and assign reviewers to papers. It is possible for some authors to be reviewers. In such a case, the reviewer should not be assigned to a paper in which he or she is an author. You should also create data for the individual reviews for each paper. The values for various rankings must be between 1 and 10.
   c. Using the MySQL loader command, load the data created in (b) into the database.
   d. Write SQL queries for the following and execute then in a MySQL interactive session.
      i. Retrieve the email and names of authors who have submitted more than one paper for review.
      ii. Retrieve the email id of the contact author, the title of paper submitted by that author, and the average rating received on its reviews such that the average rating was above 8.0.
      iii. Retrieve the paper id, title, and average ratings received for all papers sorted in decreasing order of average rating. Do not show any papers below an average rating of 3.0.
      iv. Retrieve the email and names of reviewers along with the average ratings they gave to papers assigned to them. Sort the result in increasing order of average rating.
      v. Retrieve the paper id and title of papers along with the author name and email id such that the author is also a reviewer.

2. Consider the SQL schema generated for the `GRADE_BOOK` database in Laboratory Exercise 8.28 for which the relational database was created in Laboratory Exercise 9.13.

   a. Create the database tables in MySQL.
   b. Create comma separated data files containing data for at least 20 students and 3 courses with an average of 8 students in each course. Create data for 3 to 4 grading components for each course. Then, create data to assign points to each student in every course they are enrolled in for each of the grading components.
   c. Using the MySQL loader command, load the data created in (b) into the database.
   d. Write SQL queries for the following and execute them in a MySQL interactive session.
      i. Retrieve the course numbers, titles, and term of courses in which student with id=1234 has enrolled.
      ii. Retrieve the term, section number, and course numbers along with their total enrollments.
      iii. Retrieve the term, section number, and course numbers of courses in which there is a grading component that has a weight of more than 50 and has fewer than 10 students.
      iv. Retrieve the names of students who have enrolled in all courses taught by the instructor in the Fall 2005 term.

3. Consider the SQL schema generated for the `ONLINE_AUCTION` database in Laboratory Exercise 8.29 for which the relational database was created and populated in Laboratory Exercise 9.14.
   a. Create the database tables in MySQL.
   b. Create comma separated data files containing data for at least 20 items with at least 10 buyers and at least 10 sellers. Observe the queries below and make sure that your data will be appropriate for those queries.
   c. Using the MySQL loader command, load the data created in (b) into the database.
   d. Write SQL queries for the following and execute then in a MySQL interactive session.
      i. Retrieve item numbers and descriptions of items that are still active (i.e. bidding has not closed) along with their current bidding price.
      ii. Retrieve the item numbers and descriptions of all items in which the member with member id `abc24` is the winner.
      iii. Retrieve the member ids and names of members who have an average rating of 8.0 or above.
      iv. Retrieve the bidding history for the item with item number `i246`. The bidding history should contain a chronological listing of all bids till the current time including the member id of the bidder and the bidding price.
      v. Retrieve the item number and titles of items that fall under the `/COMPUTER/HARDWARE/PRINTER` category and have "`HP`" in their titles and on which bidding has not yet closed sorted by time of bid closing.

4. Consider the `CONFERENCE_REVIEW` database of Laboratory Exercise 7.34 for which the relational database was created and populated in Laboratory Exercise 9.12. There are three types of users of the system: *contact author*, *reviewer*, and *administrator*. The contact author should be able to sign in and submit a paper to the conference; the reviewer should be able to submit reviews of papers assigned to him or her; and the administrator should be

able to assign reviewers to papers (at least three reviewers to each paper) as well as print a report of all papers sorted by average rating assigned by reviewers. The reviewers and administrator have pre-assigned user ids and passwords, however, contact authors must register themselves first before they can start using the system. Write a PhP-based application that implements the following functions:

a. A user registration page that allows a contact author to register their email and other information with the system. They choose a password in the registration page.
b. A user login page with GUI textbox elements to accept user id and password and radio buttons to choose between contact author, reviewer, and administrator. A separate menu for each user should be displayed after authentication.
c. The contact author should be able to enter details of his paper submission and upload a file.
d. The reviewer should be able to choose one of his assigned papers and submit a review for the paper.
e. The administrator should be able to assign at least three reviewers to each paper. The administrator should also be able to display a report of all papers sorted by average rating (high to low).

5. Consider the `GRADE_BOOK` database of Laboratory Exercise 8.28 for which the relational database was created and populated in Laboratory Exercise 9.13. Write and test PhP scripts that access the MySQL database that implements the following two functions for a given course section (the course number, term, and section information will be provided as GET parameters in the URL itself):
   i. Allows the instructor to enter the points/scores for a particular grading component. The instructor should be able to choose a grading component using a pull down list.
   ii. Allows the instructor to view the scores for all components for all students in a tabular form with one row per student and one column per grading component.

6. Consider the `ONLINE_AUCTION` database of Laboratory Exercise 8.29 for which the relational database was created and populated in Laboratory Exercise 9.14 in MySQL. Write PhP scripts to implement an online auction Web site. A member may register himself/herself and choose his/her password. After that, they may sign into the system. The following functions should be implemented:
   a. A buyer should be able to search items by entering a keyword. The result of the search should be a listing of items, each with a hyper-link. Upon clicking on the hyper-link, a detailed page describing the item should be displayed. On this page, the user may place a bid on the item.
   b. A seller should be able to place an item for sale by providing all details of the item along with auction deadlines etc.
   c. Buyers and sellers should be able to look at a list of all items on which bidding has ended.
   d. Buyers and sellers should be able to place a rating as well as view ratings on transactions they are involved in.