

PHP Project

Introduction

You are to write a Staff Directory web application in this project. The project is categorized into 2 level of increasing difficulties – *Easy* and *Tough*

You are only required to finish the first level, *Easy*. Some of the *Tough* parts are not tough at all. You may want to attempt that as well.

Description

Staff Directory allows a staff and general user to basically do the following:

- See a list of staff name
- Clicking on any staff name on the list will display staff profile in detail

There are three type of users.

- Admin
- Staff
- General public

Different functionalities and rights are given to each individual type:

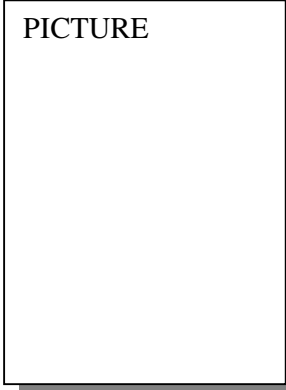
- Admin
 - Can add user
 - Can delete user
- Staff
 - Can edit one own rofile
- General public
 - Can view the list
 - Can view individual profile

The list may look like this:

John Doe	61234567	john@doe.com
Jane Doe	67654321	jane@doe.com

The name is “hyperlinked” so that when it is clicked it will display a complete profile of the person.

The page may look like this:

<p>John Doe Phone: 61234567 Email: john@doe.com Dept: Accounting City: Singapore</p> <p>Interests: Do nothing, eating</p>	
--	--

Scope

The scopes are divided into 2 level – *Easy* and *Tough*. You are only required to finish the *Easy* part.

Scope for Easy

The scope will have the following requirements:

Add User

1. A user named “admin” will have the right to ADD user
 - a. Use “admin” to add as many users as you want
2. “admin” user must login first before can ADD a user
3. Adding a user must include the following data
 - a. Full name
 - b. Login
 - c. Password
 - d. Telephone
 - e. Email
 - f. Interest
 - g. Staff’s picture
4. The picture must be uploaded and stored a proper location for display later

List User

5. Add a functionality called LIST
 - a. You don’t need to login to use this function
6. LIST all the user like the given sample above (in Description section)
7. Upon clicking on one the hyperlink, display the full profile

Tough Scope

Add the following functionalities on top of the *Easy* scope:

Search Engine

1. Add the search engine
2. Search by full name
3. List the result of the search with the name hyperlinked
4. When the hyperlinked name is clicked, you will get the full profile (just like *Easy* scope)

Delete Function

5. Add delete function
6. List all the names with “checkbox” in front of each name
7. All the selected checkboxes will be deleted upon submission

Program Design

You will have to program both the server program (PHP) and database.

Database

At least the following properties are needed:

- ID – int Not Null Auto_Increment
- Name – Varchar(50)
- Login – Varchar(12)
- Password – Varchar(12)
- Telephone – Varchar(8)
- Email – Varchar(50)
- Interest – Varchar(100)
- Picture – Varchar(100)

Or you can come up with additional properties yourself. However, the one defined above must be implemented.

You may only need one table.

Program Design

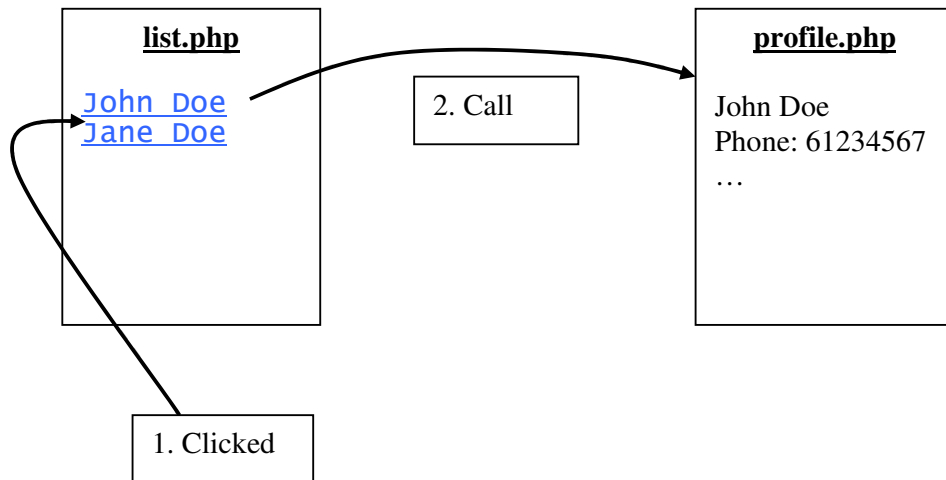
The program design should be quite straightforward. Basically you will need to track the Session data and do lots of exchanges with the database.

Passing ID

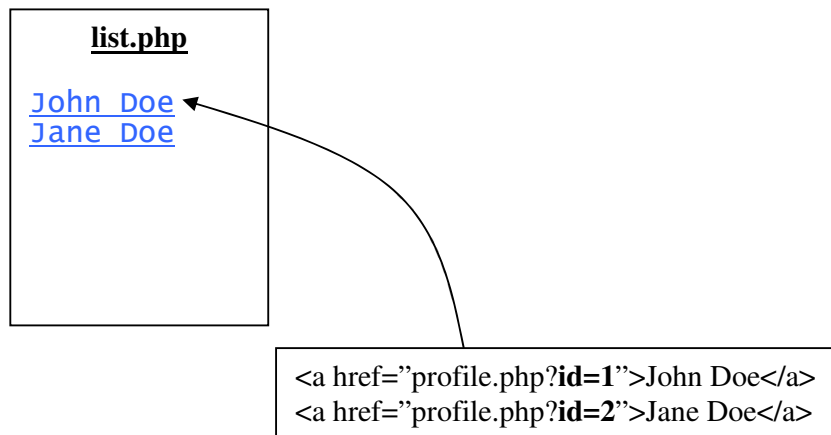
One possible difficulty is how to display the full profile from the list. Assume this list is displayed by your php program (say, “list.php”):

John Doe	61234567	john@doe.com
Jane Doe	67654321	jane@doe.com

When “John Doe” is clicked, it needs to call another application (say, “profile.php”) to actually retrieve the complete “John Doe” profile and display it.



The question is, how do you tell “profile.php” which data to retrieve. One method is to embed the ID in GET request method.



```
<a href=’profile.php?id=1’>John Doe</a>
<a href=’profile.php?id=2’>Jane Doe</a>
```

Where “id” is retrieved from the ID column in the database

The program in “profile.php” can retrieve the “id” from GET array (`$id = $_GET['id']`) and make a request one more time to the database to retrieve the row which has that “id”.

The retrieved row contains all the Profile information.

Storing Picture

The picture (or image) itself actually is not going to be stored in the database. Only the location and the name of the image file will be stored in the database.

Let say John Doe has uploaded his picture and call it “dude.jpg”. The picture is stored in “images” folder under your current folder. Then in your database you can store it like this

ID	...	Picture
1		images/dude.jpg

When the profile for John is retrieved, the picture location is also retrieved – and then displayed on the browser.

Useful References

- PHP Manual (also downloadable)
<http://www.php.net/manual/en/>
- MySQL Manual (also downloadable)
<http://www.mysql.com/documentation/index.html>
- ONLAMP web site has rich resources of PHP and MySQL article
<http://www.onlamp.com/>