COMP3322 Modern Technologies on World Wide Web

Lab 3: PHP, JavaScript and AJAX

Overview

In this lab exercise, we will develop a dynamic webpage for user login, profile update and logout, using PHP, cookie, JavaScript and AJAX technologies.

When a client accesses the web page "index.php" for the first time, the page shows up as in Fig. 1 below. The user can enter user name and password to log in.

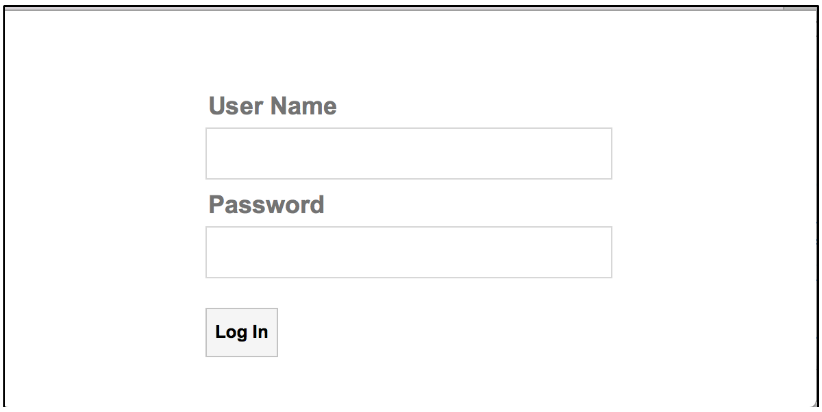
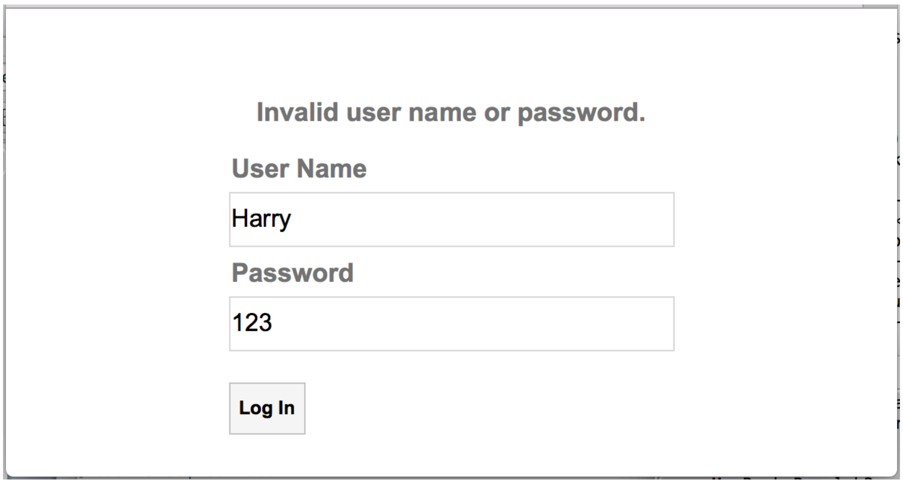


Fig. 1 Login Page.

If the user enters incorrect user name or password, the user sees a page as in Fig. 2.


Invalid user name or password.

User Name

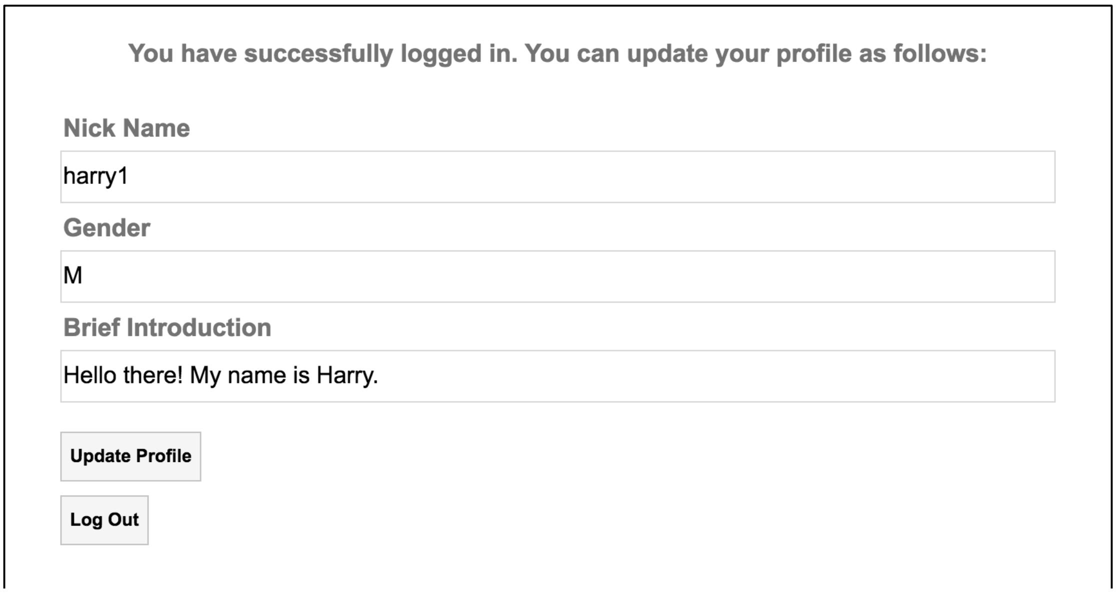
Harry

Password

123

Fig. 2 After entering invalid user name or password.

After successful login, the user sees a profile page as in Fig. 3. The three input textboxes contain the user's current profile information.



You have successfully logged in. You can update your profile as follows:

Nick Name

harry1

Gender

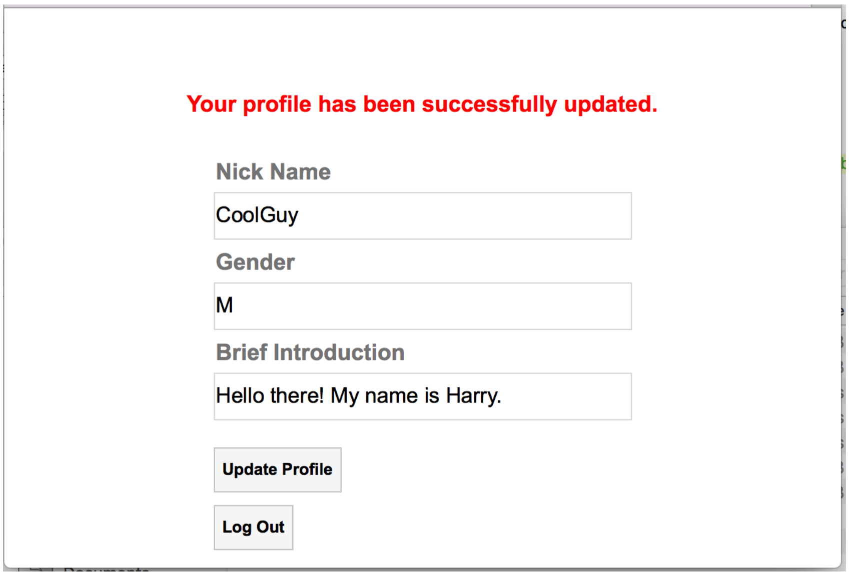
M

Brief Introduction

Hello there! My name is Harry.

Fig. 3 After successful login.

The user can then update the profile information on this page. By clicking the “Update Profile” button, the updated profile information will be sent to the server to store in the database. When the update is successful, the page is changed as shown in Fig. 4.



Your profile has been successfully updated.

Nick Name

CoolGuy

Gender

M

Brief Introduction

Hello there! My name is Harry.

Fig. 4 After successful profile update.

The user can click the “Log Out” button to log out, which will lead to the login page as shown in Fig. 1 again.

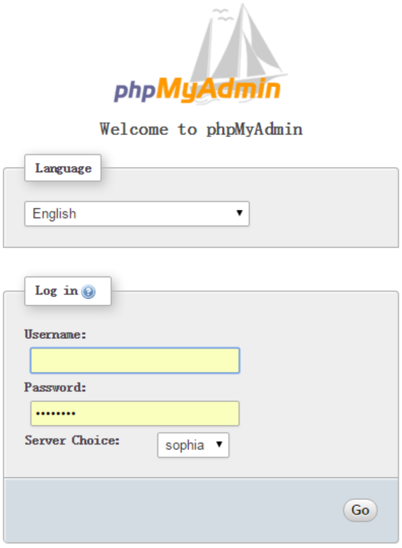
To start the lab practice, download "**lab3_materials.zip**" from HKU Moodle, and extract it to a folder named "**lab3**", which contains template files you will use to implement this dynamic webpage.

Lab Exercise 1: Set up the Database

User profiles are permanently stored in a database. So the first thing that we need to do is to set up the database for storing user profile.

[Step 1] Go to https://intranet.cs.hku.hk/common/mysqlacct/register.php and register a MySQL account, if you have not done so. It takes about one working day for CS technical staff to activate your account.

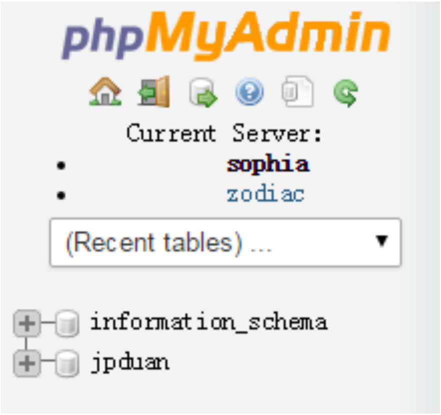
[Step 2] Go to https://i.cs.hku.hk/phpmyadmin/. Log in using your MySQL account. Choose the server **Sophia**. We will use the MySQL database hosted on sophia.cs.hku.hk.



Language: English

Server Choice: sophia

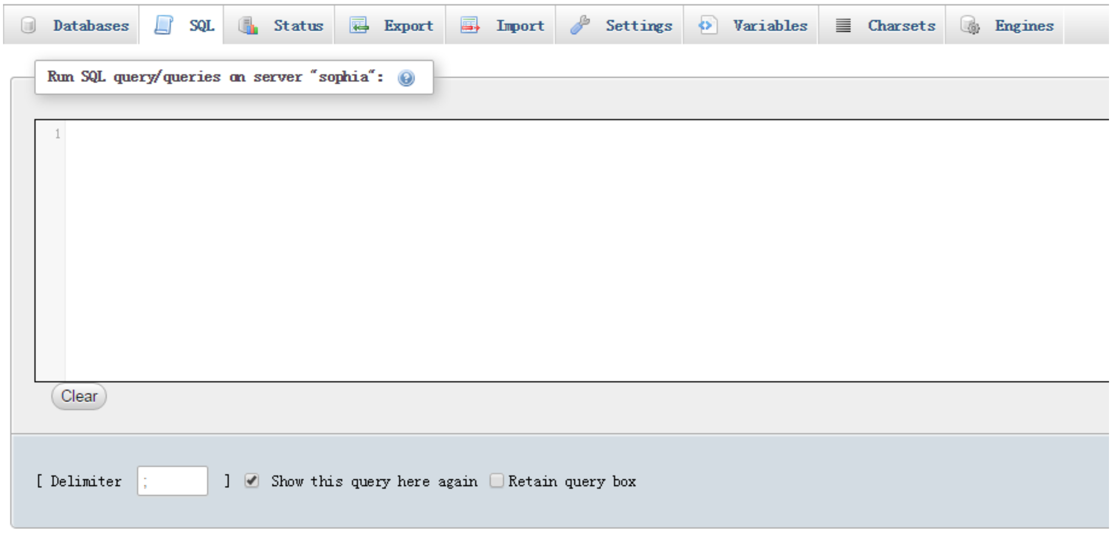
[Step 3] After logging in, select your database on the left column. Your database name should be the same as your account name.



Current Server:

- sophia
- zodiac

[Step 4] Click the SQL tab and you will see an input field for executing SQL code.



Run SQL query/queries on server "sophia":

[Delimiter ;] ☑ Show this query here again □ Retain query box

[Step 5] Create **users** table by executing the following SQL code. The **users** table contains the **userName** field and the **password** field. (All SQL statements are in sql.txt in lab3_materials.zip, which you can copy and paste.)

```
CREATE TABLE users (
 userName varchar(20) NOT NULL,
 password varchar(20) NOT NULL,
 PRIMARY KEY (userName)
)

INSERT INTO users(userName, password)
VALUES (
 'Harry',
 '12345'
);

INSERT INTO users(userName, password)
VALUES (
 'Barry',
 '23456'
);
```

[Step 6] Create **profiles** table by executing the following SQL code. The **userName** field is the primary key. It uniquely identifies a user. The **nickname**, **gender** and **briefIntro** fields are user's profile information.

```
CREATE TABLE profiles (
 userName varchar(20) NOT NULL,
 nickName varchar(20) NOT NULL,
 gender varchar(1) NOT NULL,
 briefIntro longtext,
 PRIMARY KEY (userName)
)
```

```
INSERT INTO profiles(userName, nickName, gender, briefIntro)
VALUES (
  'Harry',
  'Rock Star',
  'M',
  'Hello there!'
);

INSERT INTO profiles(userName, nickName, gender, briefIntro)
VALUES (
  'Barry',
  'Chosen 2',
  'M',
  'Hello there!'
);
```

Lab Exercise 2: Implement the Dynamic Webpage

Complete all the **TODO** items in index.php, script.js, handleLogin.php, handleUpdate.php and handleLogout.php.

[Step 1] The dynamic webpage is accessed by retrieving Index.php (e.g., by accessing **http://i.cs.hku.hk/~[YourCSID]/lab3/index.php**). In index.php, we first check whether the "username" cookie has been set: if not, index.php renders the login page as shown in Fig. 1; otherwise, it retrieves user profile information from the database and renders the profile information page as shown in Fig. 3. Please complete TODO tasks 1-5, as marked in index.php.

[Step 2] On the login page (Fig. 1), when the user clicks the "Log In" button, the "**onclick**" event will be handled by **login()** in **script.js**. **login()** generates an AJAX HTTP GET request to retrieve **handleLogin.php**. Completing TODO tasks 1-4 marked in the **login()** function in **script.js.**

[Step 3] In **handleLogin.php**, we first query the database to check whether the user's input user name and password are valid. If so, a cookie "userName" is set for the user, and the profile information page (Fig. 3) is rendered; otherwise, the page as shown in Fig. 2 is displayed. Complete TODO tasks 1-6 in **handleLogin.php.**

[Step 4] When the user updates the gender information on the profile information page (Fig. 3), an input check is performed when the "**onkeyup**" event associated with the gender input box is triggered. Complete the event handler function **inputCheck()** in **script.js**, which checks the input gender value: if the input value is not "F" nor "M" nor "" (empty), show an alert box with the message "Gender should be either F or M" and refocus the cursor on the gender input box.

[Step 5] When the user clicks the "Update Profile" button on the profile information page (Fig. 3), the "**onclick**" event will be handled by **updateProfile()** in **script.js**. **updateProfile()** generates an AJAX HTTP GET request to retrieve **handleUpdate.php**. Completing TODO tasks 1-3 marked in the **updateProfile()** function in **script.js.**

[Step 6] In **handleUpdate.php**, we update the user's profile information in the profiles table according to the values carried in the GET request. Complete TODO tasks 1-2 in **handleUpdate.php.**

[Step 7] When the user clicks the “Log Out” button on the profile information page (Fig. 3), an HTTP GET request is generated to retrieve **handleLogout.php**. In **handleLogout.php**, we unset the cookie associated with the user, and redirect the client to index.php. Complete TODO task 1 in **handleLogout.php.**

Note: You are encouraged to upload your files to i.cs.hku.hk server for testing. Pleas ensure cookie is enabled on your browser used to test the page.

Submission

Please finish this lab exercise before 23:59 Monday Oct 17. Upload the following files to i.cs.hku.hk web server under **public_html/lab3**:

index.php, handleLogin.php, handleLogout.php, handleUpdate.php, script.js, style.css

Please make sure that http://i.cs.hku.hk/~[YourCSID]/lab3/index.php is accessible.