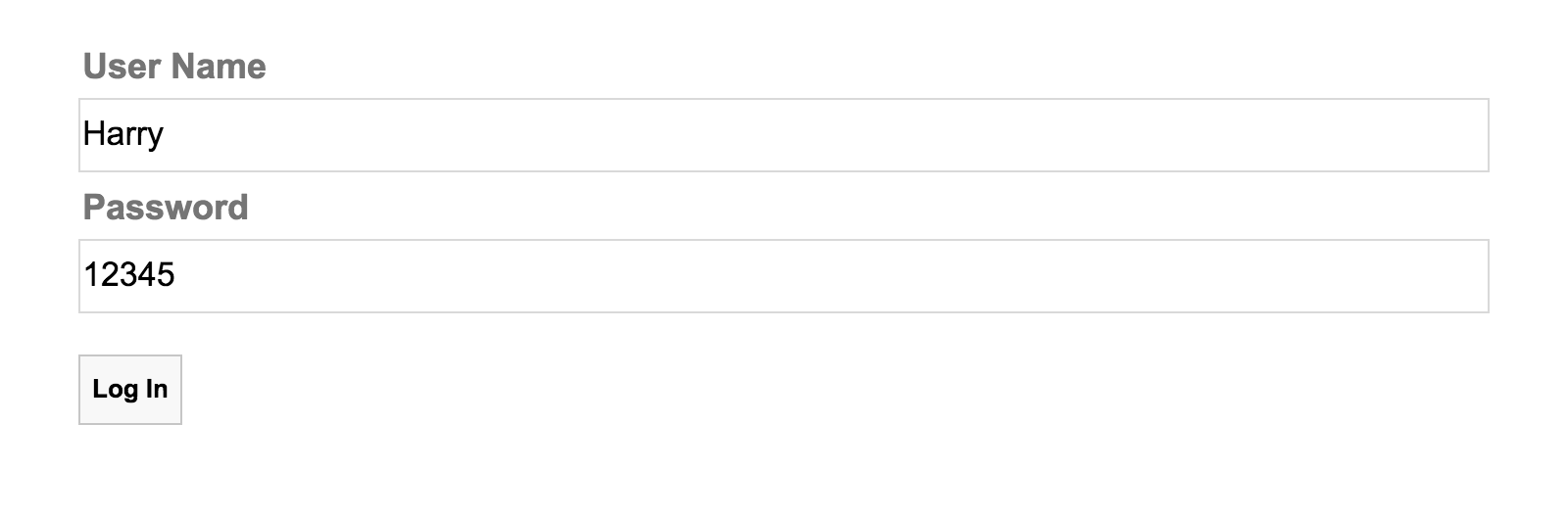
**COMP3322 Modern Technologies on World Wide Web**

**Lab 3: PHP, JavaScript, Ajax, Database**

1. Introduction

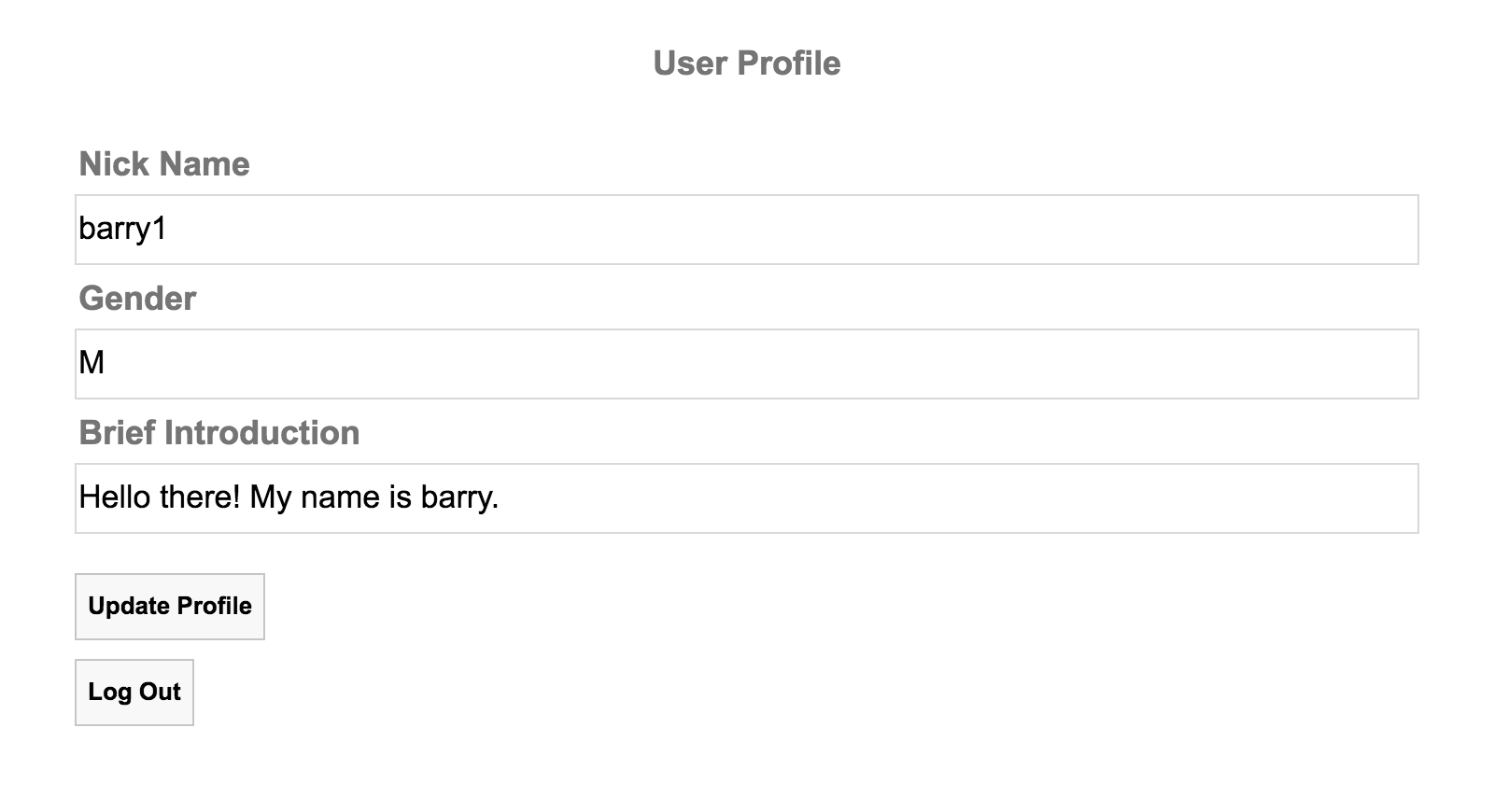
In this lab exercise, we will develop a dynamic webpage to change a user’s profile.

**1.1.** When a user enters “http://xxxx/index.html” for the first time, the user sees fig. 1. The user can enter his user name and password to log in to system.



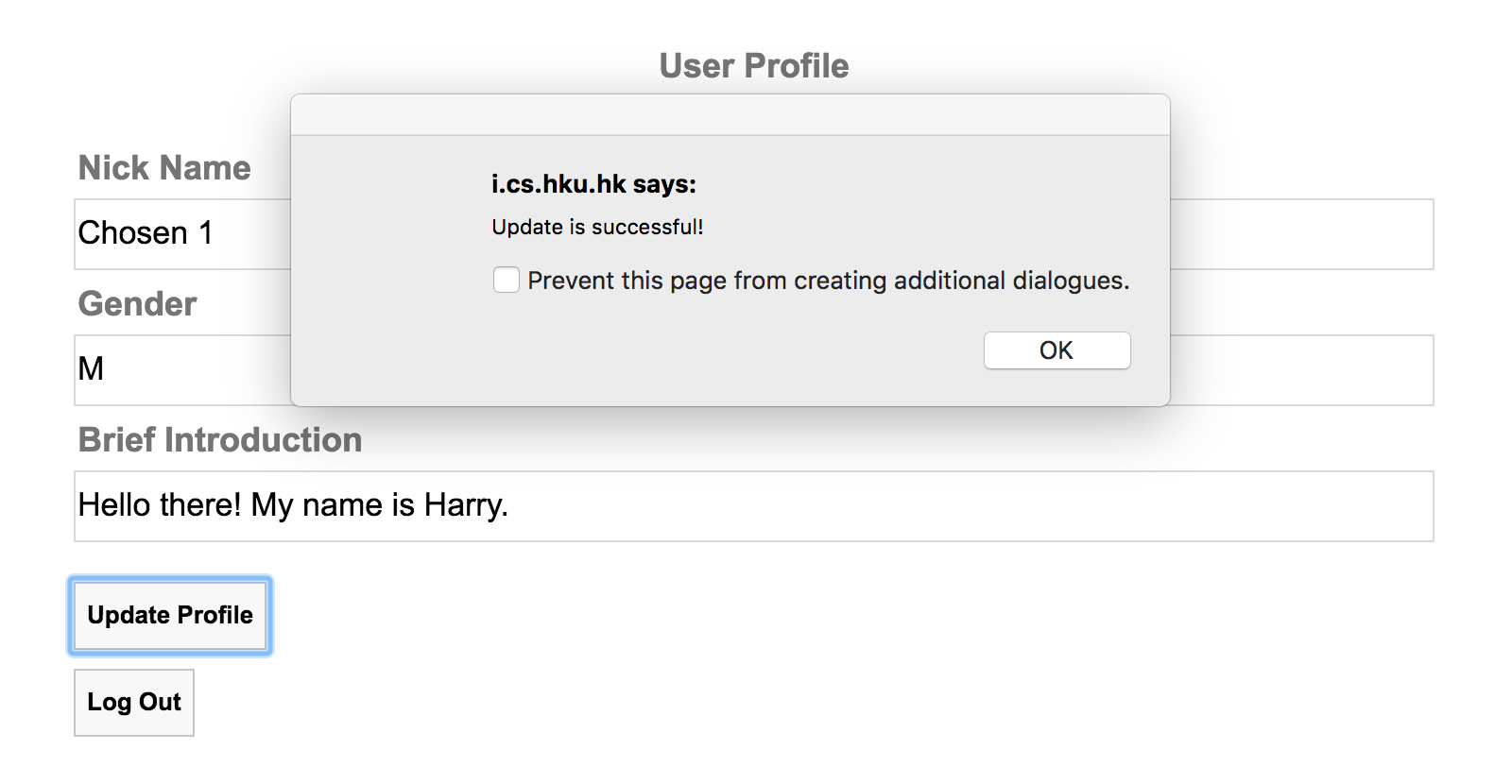
**Fig. 1 Enter user name and password to log in.**

**1.2.** After successful log in, the user sees fig. 2. The 3 input elements contain the user’s old profile values.

****

**Fig. 2 After log in.**

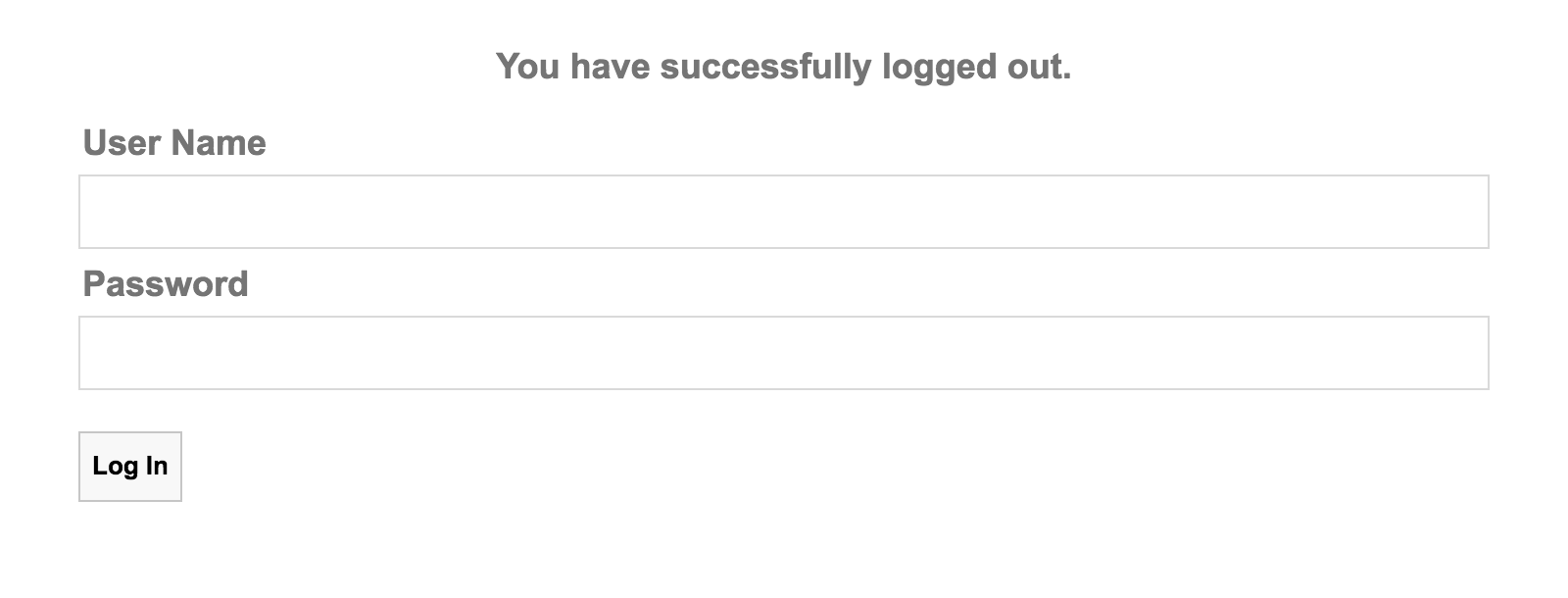
**1.3.** The user can change his profile. Then he can persist the change to the database by clicking the update profile button. When the update completes, the user receives a notification as shown in fig. 3.



**Fig. 2 Update profile.**

**1.4.** Once the user logs in, the server side uses cookie to keep track of the status of the user. The user doesn’t need to log in again even if he restarts the web page.

**1.5.** The user can click the log out button in fig. 2 to log out. After log out, the user sees fig. 4. The server side resets the user’s cookie after log out, so the user needs to log in again if he wants to change his profile later.

****

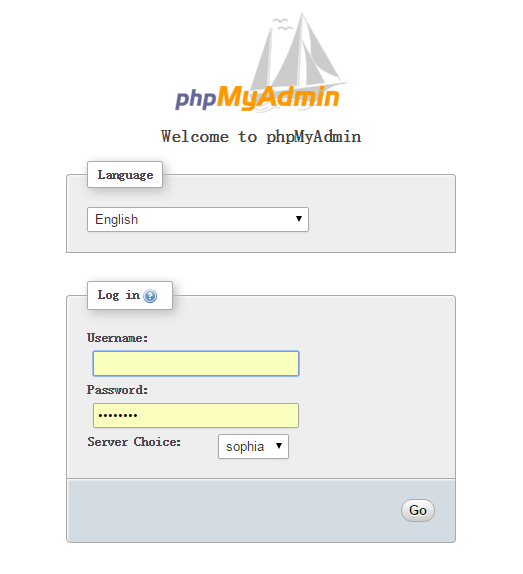
**Fig. 4 Log out.**

2. Set up the database

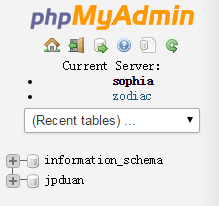
User profile is permanently saved on the database. So the first thing that we need to do is to set up the database for storing user profile.

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

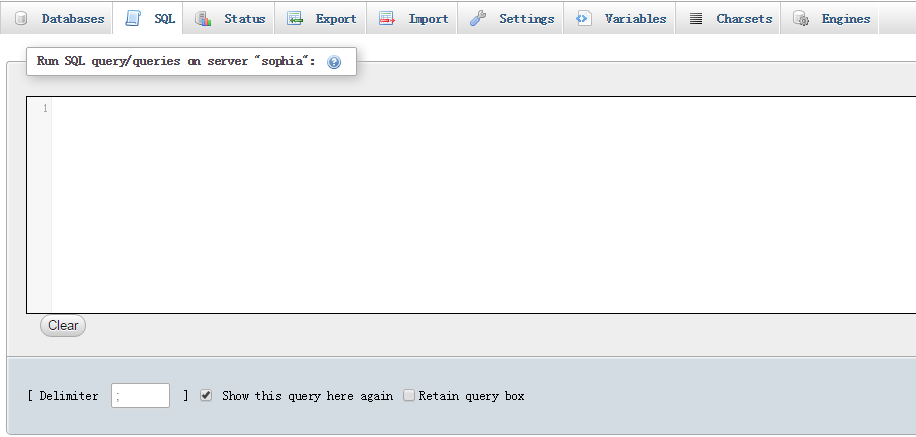
**2.2.** Go to https://i.cs.hku.hk/phpmyadmin/. Log in using your MySQL account. Choose the server **Sophia**. We will use the MySQL database hosted on sophia.cs.hku.hk.



**2.3.** After logging in, select your database on the left column. Your database name should be the same as your account name.



**2.4.** Click the SQL tab and you will see an area for executing SQL code.



**2.5.** Create **users** table by executing the following SQL code. The **users** table contains the **userName** field and **password** field. The **users** table is used to handle log in. (You can copy the code from the sql.txt contained in the lab material.)

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'

);

**2.5.** 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 that could be dynamically modified. (You can copy the code from the sql.txt contained in the lab material.)

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!'

);

**3. Complete the lab material**

Download lab3 material from Moodle and complete all the **TODO** items in the lab3 material.

**3.1.** We enter the web site using the following address “http://xxxx/index.html”. The index.html file is the entry point of the web site. Open index.html, we can see that when index.html is loaded by the browser, it calls **start()** function to render the entire webpage.

**TODO:** Please implement the **start()** function in **script.js**.

**3.2**. In fig. 1, when the user clicks the log in button, the “**onclick**” event will be handled by the **login()** function in script.js file. **login()** function generates an HTTP GET method to **handleLogin.php**. The get method should contain the input user name and password as parameters.

**TODO**: Please implement the **login()** function in **script.js**.

**3.3.** The **handleLogin.php** file on the server side handles all the login procedure. It first checks whether the user has logged in before. If so, the user is directly directed to fig. 2. If not, **handleLogin.php** file checks whether the user’s input user name and password are valid. If user’s input is valid, the user is directed to fig. 2. If user’s input is invalid, the user is directed back to fig. 1 again.

**TODO**: Implement all the missing part of **handleLogin.php**. There are totally 11 TODO items in **handleLogin.php**. You should read the comments to understand what you need to do.

**3.4**. In fig. 3, when the user clicks the update profile button, the “**onclick**” event will be handled by the **updateProfile()** function in script.js. The **updateProfile()** function generates an HTTP GET method and sends it to **handleUpdate.php**. The GET method contains the updated nick name, gender and brief introduction as parameters.

**TODO**: Please implement the client-side **updateProfile()** function in **script.js**.

**3.5.** On the server side, when **handleUpdate.php** receives the HTTP GET method, it should update the profiles table with the updated values contained in the GET method.

**TODO**: Implement the missing part of **handleUpdate.php**.

**3.6.** In fig. 3, when the user clicks the log out button, the “**onclick**” event will be handled by the **logout()** function in script.js. The **logout()** function generates an HTTP GET method and sends it to **handleLogout.php**.

**TODO**: Please implement the **logout()** function in **script.js**.

**3.7.** On the server side, when **handleLogout.php** receives the HTTP GET method, it resets the cookie associated with the user. Then it should send an HTTP response text to render fig. 4.

**TODO**: Please finish the missing part of **handleLogout.php** file.

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.html, handleLogin.php, handleLogout.php, handleUpdate.php, script.js, style.css**

**Please make sure that** [**http://i.cs.hku.hk/~[YourCSID]/lab4/index.html**](http://i.cs.hku.hk/~%5bCSID%5d/lab4/index.html) **is accessible.**