Part 1: Make the database connection

Write a PHP file that can be added to other PHP files using the include or require

functions. This file should:

• Make a connection to a MySQL database, and log in with valid credentials. The

connection resource should be stored in a variable with an appropriate name.

• Create a database group[NO] if it does not exist.

• Create a table group[NO]users if it does not exist with the following fields:

◦ USERNAME VARCHAR(100)

◦ PASSWORD\_HASH CHAR(40)

◦ PHONE VARCHAR(10)

• The USERNAME field should be designated as UNIQUE.

• If any of these operations cause an error, stop execution and print the error message

*Note:Replace group[NO] with your group number eg group1*

Part 2: Write the registration form

*Note that all of this part should be done in the same PHP file. The script should respond differently depending on the situation (whether a POST request exists, whether the username is already taken, etc.).*

Write a PHP file that will output a form containing 3 fields: username, password, and phone

number. These fields should be sent via POST to the same file, which should take care of inserting them into a database named group[NO] and table named group[NO]users (as shown above), and then confirm the registration by displaying the username and phone number back to the browser.

If the username already exists, your INSERT query should fail if you designated the USERNAME field as unique. You should query the database before attempting the insert, and if the username exists already, display an error message and a blank registration form again.

Part 3: Write the login form

Write a PHP file that will output a form containing 2 fields: username and password. Upon submission of the form, the code should check against the database to see whether the username-password pair was correct. If so, display a welcome message. If not, display the message “Invalid username or password” followed by the same login form.

Once again, there should only be one PHP file, and you should redirect to the same place after submitting. The output should be one of three options:

1. The login form.

2. The welcome message, if successful login.

3. The invalid message and the login form, if failed login.

Part 4: Write the reset password form

Write a form to allow a user to reset their password with their username and phone number. If the username and phone number match entries in the database, you should generate a random string as a password (make it of reasonable length, alphanumeric), inform the user of the random string generated, and make appropriate changes in the database. Remember that the database stores the hashed version of the password.

If the username and phone number combination does not exist, inform the user of the failure to reset the password, and display the reset form below the message.

Part 5: Write the change password form

Write a form to allow a user to change their password. It should take 3 fields: the username,current password, and new password. When the form is submitted, the code should make appropriate checks against the database, and if the username and password are correct, modify the entry to reflect the change in password. The user should receive a message that informs them of the successful password change. Remember that the database stores the hash of the password.

If the username and current password combination is not correct, the password should not be changed. Inform the user of the failure to change the password, and display the change password form below the message.