

2019 Computer Network Final Project cnMessage Report

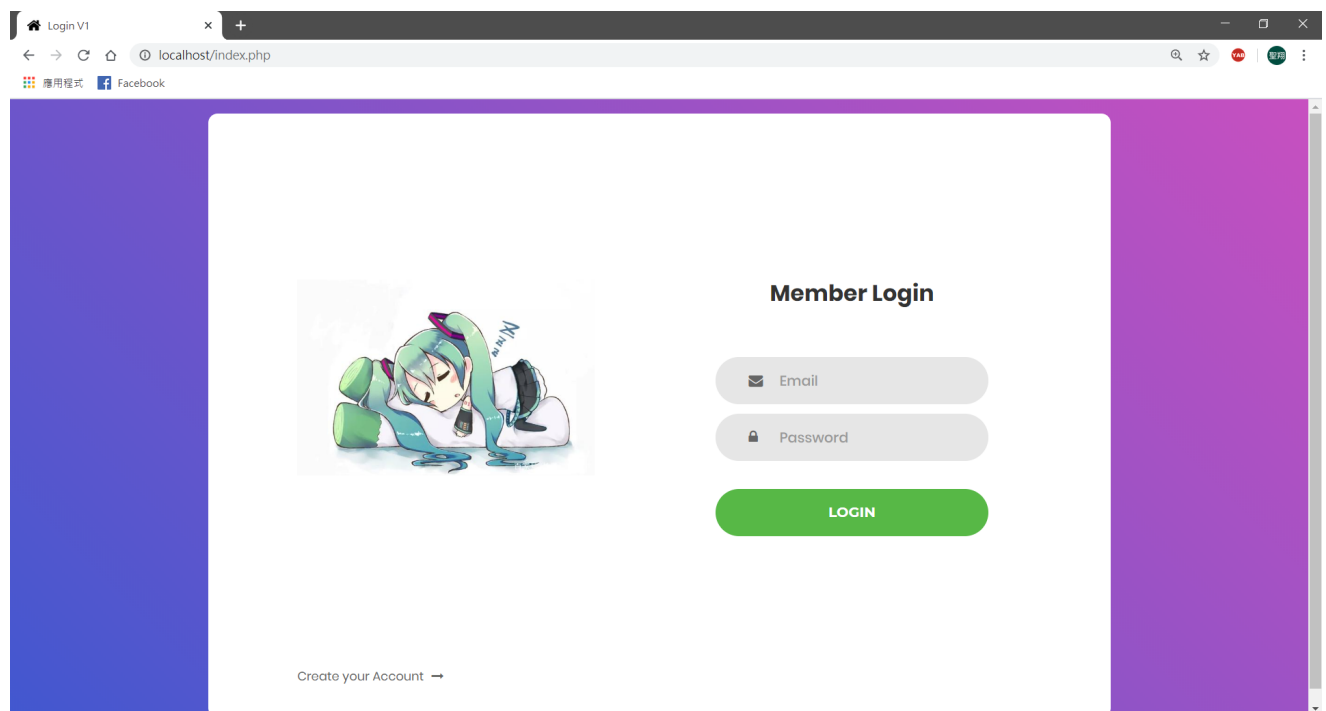
Group Member

- B06902011 陳義誠
- B06902103 尹聖翔

User & Operator Guide

- **login.php**

The login page of the website, type email and password to login. Wrong password or account will cause an error message. Jump to sign up page by clicking "Create your Account" below.



- **sign_up.php**

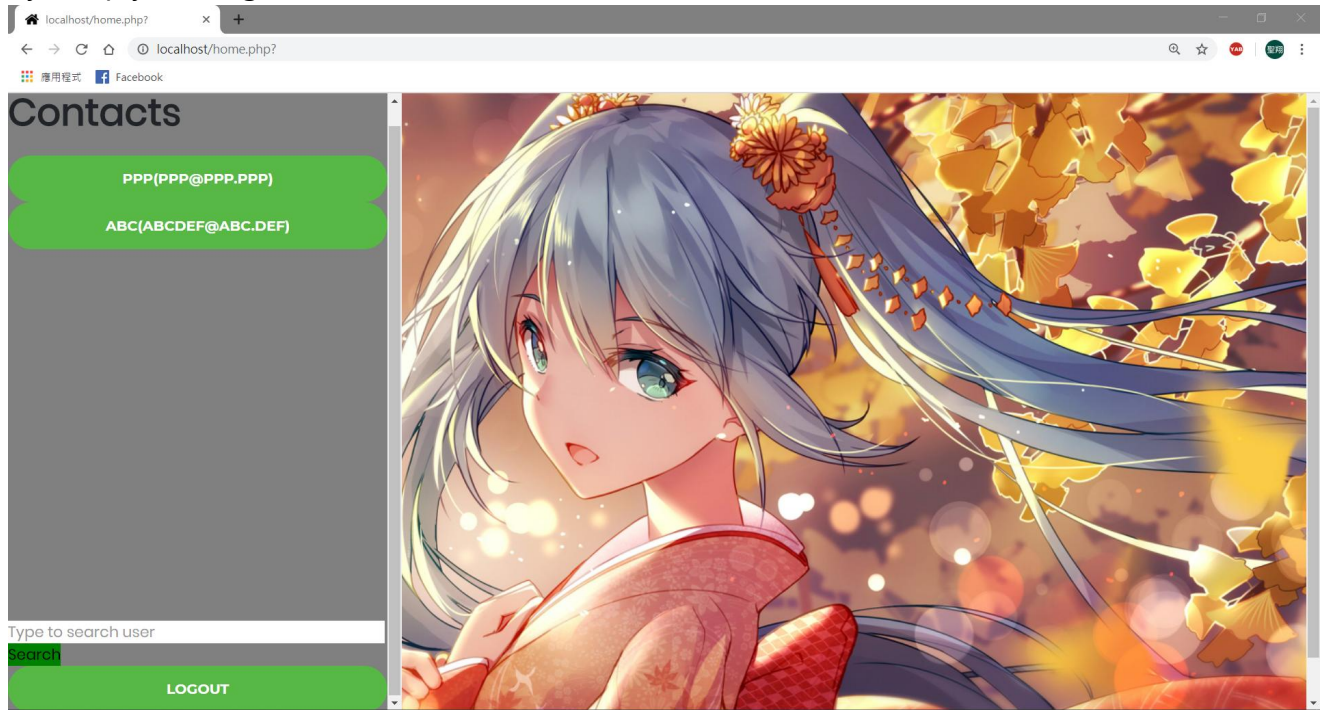
Type email, password and account to create account. Used email address is not allowed to create an account. Account information is stored in database with password encrypted.

```
mysql> select * from account_info;
+----+-----+-----+-----+
| id | email      | password | account |
+----+-----+-----+-----+
| 1  | 123456@123.456 | $2y$10$m6x0w.naKHS2SFTmI7vaEOl6PlsWWy4XMh.XrkQDvQR/nHce8Us9q | 123 |
| 2  | ppp@ppp.ppp   | $2y$10$bey3vIA1jZxCZnN4mDeGG./n9tW9BD0hJ2jUkf0020/iNgR6hhylu | ppp |
| 3  | abcdef@abc.def | $2y$10$j2m0tz8WtxJXA0hpacrHOeYlKpKtzGT9LleyUXda2wCmvrPK1n0me | abc |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

- **home.php**

The homepage showed after login. There is a list of existing chatrooms which can be pressed and enter the charoom. If you want to start a new conversation, type the account in the input box below. The chatroom will be established if the account exists. You'll need

to use the logout button to logout, auto reconnection will occur if you leave the website by simply closing the window.



- **chatroom.php**

The upper section will show all the dialogue between two clients with refreshment per one second. The lower section has two input box, one can write input conversation into database and one can choose a file to upload. When a file is uploaded, the sender will issue a hyperlink for others to download the file. Note that if you send a existing file, the

former link will be set to invalid.



Instructions on how to run Server & Clients

Server

We use Apache 2.4 to set up the server. Apache for Windows 10 does not include PHP support, so we set up by our own. Besides, the information of the accounts and the chat logs are stored in a database, so we also setup MySQL.

The server setup is under Windows 10 enviroment.

Apache Server

- Download Apache 2.4.41 for Windows from <http://httpd.apache.org> (<http://httpd.apache.org>) (we use ApacheHaus).
- After downloading and unzipping, find (path of Apache directory, which ends with "Apache24")\conf\httpd.conf and modify the SRVROOT variable as the correct path:

```
39 Define SRVROOT "D:\Apache24"
```

- Execute cmd as administrator, go to Apache24\bin, and type the command following to setup the server:

```
httpd -k install
net start apache2.4
```

- Add files in src folder we provide (src itself excluded) into Apache24\htdocs.

PHP

- Download PHP package from <https://windows.php.net/download/> (<https://windows.php.net/download/>) (We use PHP7.4.1, zip file of VC15 x64 Thread Safe.)
- Rename php.ini-development as php.ini in the PHP directory, and modify some variables in it:

```
755 extension_dir = "D:\php-7.4.1-src\ext" ((path of PHP
directory)\ext)
```

```
910 extension=php_curl.dll
```

```
913 extension=php_gd2.dll
```

```
919 extension=php_mbstring.dll
```

```
921 extension=php_mysqli.dll
```

```
926 extension=php_pdo_mysql.dll
```

- Open Apache24\conf\httpd.conf and add:

```
PHPIniDir "D:\php-7.4.1-src"
LoadModule php7_module "D:\php-7.4.1-src\php7apache2_4.dll"
<IfModule php7_module>
AddType application/x-httpd-php .php
AddType application/x-httpd-php .php3
AddType application/x-httpd-php .php5
AddType application/x-httpd-php .php7
</IfModule>
```

MySQL

- Download MySQL Community Server from <https://dev.mysql.com/downloads/mysql/> (<https://dev.mysql.com/downloads/mysql/>) (Windows (x86, 64-bit), ZIP Archive)

- Add new file my.ini under the MySQL directory with the following content:

```
1 [mysql]
2 default-character-set=utf8
3 [mysqld]
4 port = 3306
5 basedir=D:\\mysql-8.0.18-winx64
6 max_connections=200
7 character-set-server=utf8
8 default-storage-engine=INNODB
```

- Execute cmd as administrator, go to (path of MySQL directory)\\bin, and type:

```
mysqld -install
mysqld --initialize
net start mysql
```

- Type the following command to login as root. The password is stored in mysql-8.0.18-winx64\\data:

```
2020-01-11T08:32:41.859990Z 5 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: d+5FOum/nttF
```

```
mysql -u root -p
```

- In case the error showed in the following happens, you could use the command:

```
mysql -u root -p --default-character-set=utf8
```

```
mysql: Character set 'utf8?' is not a compiled character set and is not specified
in the 'C:\Program Files\MySQL\MySQL Server 8.0\share\charsets\Index.xml' file
```

- After logging in, do the following query:

```
CREATE DATABASE userDB;
use userDB
CREATE TABLE account_info(id INT PRIMARY KEY AUTO_INCREMENT, email
VARCHAR(255), password VARCHAR(255), account VARCHAR(255),
UNIQUE(email));
```

- Our codes, which would connect to localhost mySQL, passes the password "root". Therefore the password should be changed with the following command (in MYSQL):

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'root';
```

Clients

Just simply type the ip on the web browser and you can connect to the server :) (Assume the server has a public ip.)

System & Program Design

Registration

- File: **sign_up.php** & **check_sign_up.php**
sign_up.php requires user their email, password and account name. Then it will submit them via POST to **check_sign_up.php**, which would query the database to check whether the email has been registered. On success, **check_sign_up.php** would display "Successfully signed up.", and the information of the account would be stored in the database. The password value stored is hashed so that others cannot get the password even if they have access to the database for some reason. On failure, **check_sign_up.php** would display the error (ex. email already used). On **check_sign_up.php** there is a return button redirecting to **index.php** to login.

Login

- File: **index.php** & **login_check.php**
index.php collect input(email and password) and then send them to **login_check.php** via POST method to exam whether the email is in database(table name=account_info). The password is encrypted by password_hash function of PHP and examined by password_verify. When login succeed, there will be some PHP session variables declared. Those variables can be used to exam whether a user had logged in, and a user who had already logged in will be redirected to **home.php**.

Unauthorized Users Prohibition

- File: **home.php** & **chatroom.php**
The two pages, which are the pages clients can mainly access to, put check on whether the users are actually login through **index.php**. The check is implemented by checking if the PHP session variables are declared. If not, the window is going to be redirected to **index.php**.

Search for New User

- File: **home.php** & **search.php**
There is a search bar under the left bottom of **home.php**. User can use email or account name to search. Once clicking the search button, it would POST the search content to **search.php**, which queries the database to find if there are matches between the search content and the email, or the account name. Then, it would list all the matching results as buttons, clicking which would redirect to **chatroom.php** to chat. If there are no matches,

"No results." is displayed. On **search.php** there is a return button to redirect to **home.php**.

Contact List

- File: **home.php**

Once the user chats with another, the database would create a table to store their chat log. Therefore, on loading **home.php**, it would query the database to check if there is a chat log table between the user and another. If there is, a button would be displayed on the left side of **home.php** with the other one's email and account name so that the user can click it to chat immediately without searching again. If no such table exists, "No contacts." would be displayed on the left side of **home.php**.

Chat

- File: **chatroom.php** & **send_context.php** & **get_content.php**

We use XMLHttpRequest to achieve refreshment of conversation only without reload the whole page. When a conversation begun, there'll be a table named by the speakers established to store every message. **send_context.php** will write the input dialogue as well as the speaker's id and issue time into that table and **get_content.php** will search for new record of dialogue according to the timestamp. The content will check per one second, however, every message sent will trigger one content check.

File Upload

- File: **chatroom.php** & **upload.php**

We use PHP upload via POST to achieve file upload function requiring change the PHP setting related to file upload and post max size. The uploaded file will be store in a folder, named after two speakers, in htdocs. Every time a file is sent, a hyperlink will be write into the dialogue table so that one can download the file through the link showed in the chatroom. Whether a file is already existed will be check, if existed, the older link will be automatically invalid.

Logout

- File: **home.php** & **chatroom.php** & **logout.php**

There is a logout button for user to logout on **home.php** and **chatroom.php**, which would redirect to **logout.php**. **logout.php** would destroy the session and redirect to **index.php**, and the user is logged out.

Others

Bonus

- Contacts list: described above.
- Duplicate file link invalid
When a file is uploaded, there is a hyperlink for it in the dialogue log. However, if a user uploads a file with the same filename, the previous hyperlink is now linked to the new file, which may cause misleading. Therefore, on detecting this the **chatroom.php** would modify the chat content immediately and the previous hyperlink becomes the non-link plain text "filename (Invalid: expired or replaced)".
- Auto reconnection
We use session so that even if the webpage is closed, the user can use its account without logging in again when reopen the page.
- Special character escape for chat
Since the display of the chat log is of HTML form, some characters would be displayed wrongly if no special handling is applied. For example, If you send "
" to another, the display would be a newline. Sending "<" would be displayed as "<". To avoid this, we replace "&" with "&", "<" with "<", and ">" with ">". We also add "\" before "'" (apostrophe) and "\". Now our chatroom can display "&", "<", ">", "\", "'", correctly.
- Avoid repeated display
When the time user send and request chat log is very closed to that of auto update, repeated contents would be displayed. We make **chatroom.php** able to remove these contents. Nevertheless, sometimes it would not work if you send the message too frequently.
- Unauthorized users prohibition: described above.

Setup Reference

- The Apache server and PHP set up is referenced to <https://www.itread01.com/content/1548943214.html>
(<https://www.itread01.com/content/1548943214.html>)
- MySQL setup is referenced to <https://blog.csdn.net/yulutian/article/details/78971369>
(<https://blog.csdn.net/yulutian/article/details/78971369>)